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. We show that the variational objectives used by existing explainers are invariant to the joint structure of the selected subgraph, and that the module's importance-scaled noise provably reduces input dependence while bounding predictive leakage. To address this, we propose BottleneckMLP, a model-agnostic module inserted between the GNN encoder and the explainer head consisting of an MLP compression block and a noise component. The compression block delivers an implicit information bottleneck, the first such two-phase IB dynamic observed in GNN representations, and replaces explicit variational IB losses without performance loss. By injecting Gaussian noise inversely scaled by node importance, BottleneckMLP yields embeddings where important nodes remain structured and clustered, while unimportant nodes drift toward Gaussianized, high-entropy distributions. This importance-scaled Gaussian noise component contributes a further 5-15% improvement in explanation fidelity (Fid+, Fid-), and accelerates convergence by 30-40% across datasets. Across 6+ ante-hoc and post-hoc explainers spanning graph and node classification, link prediction, subgraph recognition, and temporal graphs, BottleneckMLP recovers and exceeds existing baselines.

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. Explainability methods (Dai et al., 2024; Li et al., 2023b; Yuan et al., 2022) are essential to build trust and ensure reliability in sensitive applications such as healthcare and scientific discovery. These methods aim to find a graph explanation as a subgraph that is necessary and sufficient to explain the downstream GNN task.

Post-hoc GNN explainability methods (Bajaj et al., 2021; Baldassarre & Azizpour, 2019; Luo et al., 2020a) apply an explainer to a pre-trained black-box model. Recent work 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. BottleneckMLP integrates with post-hoc and ante-hoc explainers, and generalizes across different downstream tasks such as graph classification, node classification, and subgraph recognition.

The Information Bottleneck (IB) principle (Tishby et al., 2000; Tishby & Zaslavsky, 2015) formalizes the 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

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

dependencies make learning compact, task-relevant representations challenging. Previous GNN explainers (ante-hoc (Miao et al., 2022), post-hoc (Chen et al., 2024), prototype-based (Seo et al., 2024b)) include IB losses to encourage the learned representation to be sufficient yet minimal.

We formalize GNN explanation as an IB problem where deeper layers naturally compress X (input graph G), retaining only the information in Z (explanatory subgraph G_S) needed for predicting Y. We argue that explicitly minimizing $I(X;Z)$ via auxiliary explicit IB losses is ineffective. We reconstruct the information plane for GNNs and show consistency with IB theory (Tishby & Zaslavsky, 2015). We focus on IB-based explainers to show that ablating their explicit regularization and applying our method recover IB behavior and surpasses their original performance. We demonstrate that our framework is universally applicable by showing that it significantly improves accuracy and explanation quality in all types of explainers. Our key contributions are threefold:

(1) **Exact characterization of what explicit IB losses constrain.** We show that the IB losses in GSAT, PGIB, and TGIB decompose into a sum of per-unit terms, so they are invariant to the joint structure of the selected subgraph. Their gap to the intended objective is exactly the redundancy among selected units plus the mismatch between the aggregate posterior's marginals and the prior, both of which are reducible without reducing $I(G;G_S)$.

(2) **BottleneckMLP: An Architectural Implicit IB Module with Importance-Aware Compression.** A model-agnostic module combining two complementary pieces: (i) an MLP compression block, and (ii) importance-scaled Gaussian noise that ties compression to the explainer's importance signal. We prove that the noise component monotonically reduces $I(X;Z_{\text{unimp}})$, that label-relevant leakage through unimportant nodes is bounded by that same term, and that retained information is graded by importance. Empirically the noise component contributes 5-15% to explanation fidelity (Fid+, Fid$-$) and accelerates convergence by 30-40%, while the MLP block supplies most of the accuracy gain.

(3) **Empirical Validation.** Across ante-hoc and post-hoc explainers for graph, node classification, and subgraph recognition, BottleneckMLP improves explanation quality by up to 150% and performance by up to 42%. Representation dynamics further indicate that it induces IB behavior not observed in baseline models.

2 Preliminaries

Mutual information (MI) measures how much one random variable reduces the uncertainty of another: $I(X;Z) = H(X) - H(X|Z)$, where $H(\cdot)$ denotes entropy. We use EDGE (Noshad et al., 2018) as the MI estimator throughout. A complete list of symbols is in Appx. A.

Information Bottleneck (IB) (Tishby et al., 2000) learns a minimal sufficient representation Z that retains only the aspects of X relevant to Y, with β balancing relevant information against compression: $\min_{p(z|x)} \left[-I(Z;Y) + \beta I(X;Z)\right], \quad \beta \geq 0$. (Tishby & Zaslavsky, 2015) interpret DNN training via the IB framework, where (Shwartz-Ziv & Tishby, 2017) identify two distinct phases in MI dynamics. The first is the fitting phase where the model fits the data, minimizing cross-entropy increases $I(X;Z)$ and $I(Z;Y)$. The second is the compression phase, where layers discard task-irrelevant information, reducing $I(X;Z)$ while preserving or increasing $I(Z;Y)$. Compression proceeds slowly and without explicit regularization. The two-phase dynamic of DNN training emerges in 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 predictive accuracy and explanation quality.

3 Related Work

IB-Based Explainers Explanatory subgraphs vary in size, making fixed-size constraints ineffective (Kakkad et al., 2023). Information-constrained methods Achille & Soatto (2017); Alemi et al. (2018) instead use the IB principle (Tishby et al., 2000), limiting retained information rather than size. IB-based explainers such as GSAT (Miao et al., 2022), PGIB (Seo et al., 2024b), and TGIB (Seo et al., 2024a) encourage minimal sufficient representations via $\min_\phi -I(G_S;Y) +$

$\beta I(G_S; G)$, s.t. $G_S \sim g_\phi(G)$. 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 extracts subgraphs R_k from temporal neighborhoods, via an analogous IB objective. These methods rely on variational upper bounds to constrain $I(X; Z)$. We show these objectives are invariant to the joint structure of the selected subgraph. BottleneckMLP achieves the compression phase of the IB curve without IB loss terms.

Compression in Deep Learning DNNs undergo an implicit compression phase during supervised training. Motifs emerge without explicit regularization (Scabini & Bruno, 2023), and simple fully connected layers improve CNN generalization without compression losses (Basha et al., 2020; Kocsis et al., 2022), suggesting architectural biases alone can induce compact, informative representations. (Chou et al., 2025) reframe this as a lazy–rich dichotomy: a rich phase rapidly learns task-aligned features, followed by a prolonged lazy phase of slower forgetting and stabilization, with training passing through distinct stages of manifold untangling. We read these as IB phases, with rich learning increasing $I(Z; Y)$ through feature alignment and the lazy regime suppressing noise, reducing $I(X; Z)$. BottleneckMLP applies this to GNN explainability by driving architectural compression during the lazy phase without explicit IB loss terms.

Gaussianization of Representations Gaussian distributions are the most efficient signal and worst-case noise, and enforcing Gaussianity improves generalization and robustness across architectures (Eftekhari & Papyan, 2025). Extending infinite-width theory to bottleneck NNGPs, (Agrawal et al., 2020) show that bottleneck layers act as compression points preserving task-relevant information. Structured architectures therefore promote robustness and generalization on their own, supporting our view that compression and explanation need not rely on explicit IB constraints. BottleneckMLP follows this principle, achieving compression and explanation by architectural design.

4 BOTTLENECKMLP

Explainability arises from representation dynamics rather than auxiliary loss. When a model learns a minimal sufficient representation, task-relevant structure concentrates into a subset of stable nodes, while irrelevant components are progressively forgotten. We introduce a theoretically-grounded architectural mechanism that induces such selective compression.

4.1 WHAT EXPLICIT IB OBJECTIVES CONSTRAIN

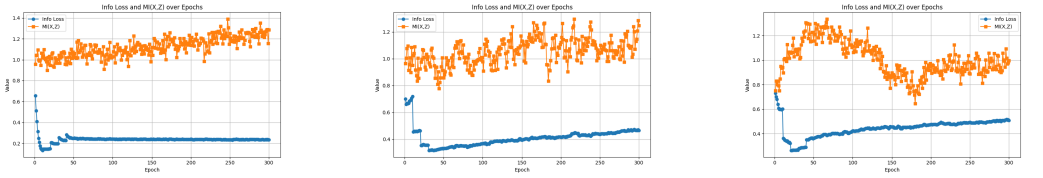
The main strategy across IB-based methods is to replace the intractable term $I(G; G_S)$ with a variational upper bound against a factorized prior. We show the gap in this bound is exactly the redundancy among selected units plus the mismatch between the aggregate posterior's marginals and the prior, and that the resulting loss is a sum of per-unit terms invariant to the joint structure of the selected subgraph. Our critique concerns the tractable instantiation these methods adopt, not the validity of variational MI bounds. Explainer-specific details are in Appx. N.

Variational Upper Bounds. The terms of the Lagrangian in the IB equation 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, so variational bounds for MI (Poole et al., 2019) are used (Du et al., 2020; Dai et al., 2018; Bao, 2021; Li et al., 2023a). Maximizing $I(Z; Y)$ reduces to cross-entropy loss: $I(Z; Y) = \mathbb{E}_{p(y,z)}[\log p(y|z)] - H(Y)$. Ante-hoc graph explainers (Lee et al., 2023; Miao et al., 2022; 2023; Seo et al., 2024a;b) constrain $I(X; Z)$ with a variational upper bound against a factorized prior. The loss is defined as: $\mathcal{L}_{VUB} := \mathbb{E}_G[\mathrm{KL}(P_\phi(G_S \mid G) \,\|\, Q(G_S))]$, where edges $\hat{e}_{ij}$ are parameterized by $\hat{e}_{ij} \sim \mathrm{Bernoulli}(\phi_{ij})$ and the prior $Q(G_S)$ by $e_{ij} \sim \mathrm{Bernoulli}(r)$. We refer to this as 'IB Loss' throughout. PGIB uses a contrastive loss (Appx. N) to minimize $I(X; Z)$, which we also call 'IB Loss'. Section 4.1 characterizes what this objective does and does not constrain.

Proposition 1 (Gap decomposition for factorized-prior IB explainers)**.** *Fix a finite unit set $\mathcal{K}$ (edges, temporal events, or nodes). Let an explainer emit a mask $\hat{s} = (\hat{s}_k)_{k\in\mathcal{K}} \in \{0, 1\}^{\mathcal{K}}$ with*

$$P_\phi(\hat{s} \mid U) = \prod_{k\in\mathcal{K}} \mathrm{Bern}\big(\phi_k(U)\big), \qquad Q = \prod_{k\in\mathcal{K}} \mathrm{Bern}(r), \qquad \phi_k(U) \in (0, 1),$$

i.e. the units are conditionally independent given the explainer input U. Write $\bar{\phi}_k := \mathbb{E}_U[\phi_k(U)]$ and let $\bar{P}_\phi$ denote the aggregate posterior of $\hat{s}$. Then by Lemma 2 (Appendix B), the explicit IB loss



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

Figure 1: Mutual information $I(X;Z)$ (orange) and IB Loss (blue) over epochs (GSAT on MUTAG). BottleneckMLP enforces compression, 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 ($I(X;Z)\downarrow$) phases consistent with (Tishby & Zaslavsky, 2015). Note that minimization of IB Loss (blue) does not correlate with $I(X;Z)$ (orange). The x-axis shows epochs 0–300. The y-axis is the loss/MI value in range (0, 1.4).

decomposes as

$$\mathcal{L}_{\mathrm{IB}} \;:=\; \mathbb{E}_U\big[\mathrm{KL}(P_\phi(\cdot \mid U)\|Q)\big] \;=\; \sum_{k\in\mathcal{K}}\Big[\underbrace{I(U;\hat{s}_k)}_{\textit{per-unit information}} + \underbrace{\mathrm{KL}\big(\mathrm{Bern}(\bar{\phi}_k)\|\mathrm{Bern}(r)\big)}_{\textit{marginal mismatch}}\Big], \tag{1}$$

and its gap relative to the intended target is exactly

$$\mathcal{L}_{\mathrm{IB}} - I(U;\hat{s}) \;=\; \underbrace{\mathrm{TC}(\bar{P}_\phi)}_{=\sum_k I(U;\hat{s}_k)-I(U;\hat{s})\,\geq\,0} + \sum_{k\in\mathcal{K}}\mathrm{KL}\big(\mathrm{Bern}(\bar{\phi}_k)\|\mathrm{Bern}(r)\big), \tag{2}$$

where TC *is the total correlation of the units under the aggregate posterior.*

For GSAT, $U = G$ and G_S is determined by $(G, \hat{s})$, so $\hat{s}$ contains the information that $I(G;G_S)$ is intended to constrain. The same follows for TGIB with R_k and for PGIB's variational branch. The explainer-specific instantiations of Proposition 1 are given in Corollaries 1–4.

The loss is a sum of per-unit terms and contains no functional of the joint law of the mask. Two explainers with identical unit marginals therefore incur the same IB loss even when one concentrates its selected units in a single connected motif and the other spreads the same information redundantly across the graph, so the objective cannot distinguish them despite very different $I(U;\hat{s})$. Equation equation 2 identifies the two directions along which the loss can be reduced without reducing the target. These are (1) de-correlating unit selections (shrinking TC) and (2) pulling the aggregate marginals toward r. Message passing inflates the first term since embeddings of units sharing a multi-hop neighborhood produce strongly dependent selection probabilities. This accounts for the decoupling in Figure 1, where $\mathcal{L}_{\mathrm{IB}}$ falls while $I(X;Z)$ does not.

Measuring the Gap. We check Proposition 1 exactly. On a synthetic explainer small enough for the joint law over masks to be enumerated in full rather than sampled ($|\mathcal{K}| = 6$, 8 inputs), both equation 1 and equation 2 hold ($|\Delta| \leq 8.3\times10^{-16}$; Appx. F). Two explainers with identical per-unit marginals but differing joint dependence incur identical loss ($\mathcal{L}_{\mathrm{IB}} = 1.096798$ for both, difference 0), validating the invariance the proposition predicts. The marginal-mismatch term is unchanged at 0.086 and the entire difference is absorbed by the total-correlation term, which rises from 0.121 to 0.384 while the target $I(U;\hat{s})$ falls from 0.890 to 0.627.

On GSAT we measure the decomposition directly from the logged attention weights, with no MI estimator involved. Table 1 reports it across five datasets. The marginal-mismatch term accounts for 10–70% of $\mathcal{L}_{VUB}$, reaching roughly 70% on AIDS and NCI1. This term is determined entirely by the dataset-average selection rate $\bar{\phi}$ and the prior rate r, so it carries no information about which edges are selected and cannot reflect compression structure. The displacement of $\bar{\phi}$ and r shows that equilibrium sits away from r, so the penalty is not a static offset but a term whose gradient opposes the prediction loss throughout training. PROTEINS appears lower due to the high $\bar{m}$.

By Corollary 2 the mismatch term certifies a lower bound on the gap of $\bar{m}\,\mathrm{KL}(\mathrm{Bern}(\bar{\phi})\|\mathrm{Bern}(r)) - H(M)$ even under variable graph size, where $H(M)$ is the entropy of the edge-count distribution. The bound is positive on four of five datasets, certifying that at least 11.8 of NCI1's 22.8 nats and

Table 1: Decomposition of the VUB in GSAT, measured from edge attention over the test set ($r = 0.5$). By Corollary 1, Mismatch $+ \bar{m}\, I_{\tilde{P}}$ sums to $\mathcal{L}_{VUB}$ exactly under the size-biased law. "Frac." is Mismatch/$\mathcal{L}_{VUB}$. "Certified" is the lower bound on the gap $\mathcal{L}_{VUB} - I(G;\hat{s})$ from Corollary 2, namely Mismatch $- H(M)$, where $H(M)$ is the entropy of the edge-count distribution.

Dataset	$\mathcal{L}_{VUB}$	Mismatch	$\bar{m}\, I_{\tilde{P}}$	Frac.	$H(M)$	Certified
AIDS	9.240	6.497	2.743	0.70	2.794	3.703
BA-2Motifs	9.867	2.544	7.323	0.26	0.921	1.623
MUTAG	11.476	4.085	7.391	0.36	3.586	0.499
NCI1	22.816	15.630	7.186	0.69	3.808	11.822
PROTEINS	29.590	2.898	26.692	0.10	4.218	−1.320

3.7 of AIDS's 9.2 cannot be information about G. It is weak on MUTAG (0.50 nats) and vacuous on PROTEINS, whose 76 graph sizes across 112 test graphs make $H(M)$ exceed the mismatch term.

4.2 BottleneckMLP Architecture

BottleneckMLP is a two-component architectural module that implicitly enforces the IB principle.

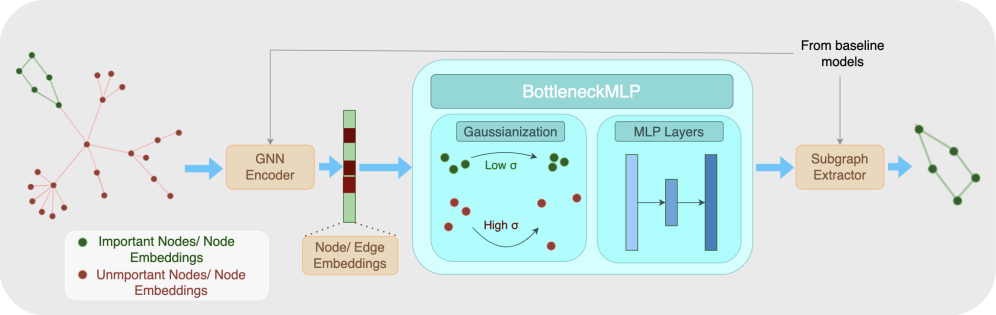


Figure 2: BottleneckMLP method for a general GNN pipeline. Our general module acts on the embeddings produced by the explainer's GNN module, and the transformed embeddings are directed to the subsequent subgraph extractor component.

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:

$$Z_i = f_i(X) + \sqrt{t_i}\,\epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, I_d), \quad t_i := \frac{\sigma}{\alpha_i},\ \alpha_i \in (0,1), \tag{3}$$

where $f_i(X) \in \mathbb{R}^d$ is the output of the encoder, and σ is a fixed hyperparameter, and the noise terms $\{\epsilon_i\}$ are sampled independently at each forward pass. This mechanism scales the variance of injected Gaussian noise inversely with importance, so that unimportant nodes are perturbed more heavily, thereby pushing them toward high-entropy representations.

Component 2: Progressive Compression via MLP Layers Following (Tishby & Zaslavsky, 2015), we employ an MLP to filter noise, compress representations, and yield compact, salient embeddings. The default configuration is $h \to \frac{h}{4} \to h$, with ReLU activations.

Integration with Existing Explainers. BottleneckMLP integrates seamlessly into existing pipelines (Figure 2), operating directly on the GNN embeddings before subgraph extraction. The expansion layer is not essential for inducing bottleneck behavior; it primarily ensures architectural modularity. In composition with the noise component, the capacity bound $I(X;\tilde{Z}) \leq \frac{1}{2}\sum_j \log(1 + \tilde{\lambda}_j/t)$ is governed by the eigenvalues $\tilde{\lambda}_j$ of $\mathrm{Cov}(\tilde{Z})$, whose participation ratio tracks the bottleneck width rather than h (proof in Appx. E; verification in Appx. F). This modular design replaces explicit IB loss terms and generalizes across different explainers and tasks (see Appx. M).

Importance Score Computation Node importance scores are derived from each explainer's native mechanism: GSAT uses stochastic attention weights, PGIB prototype similarity, and TGIB tem-

poral neighborhood relevance. This preserves compatibility while leveraging each model's built-in scoring. We defer a longer discussion of BottleneckMLP's robustness to Appx. Z.

4.2.1 IMPLICIT IB THROUGH GAUSSIANIZATION

Implicit Realization of the IB Lagrangian BottleneckMLP induces representation dynamics consistent with the IB objective. Importance-scaled Gaussian noise ties compression to the explainer's importance signal α. Theorem 1 shows it reduces $I(X; Z_{\text{unimp}})$ monotonically, strictly under conditional non-redundancy. Decomposing $I(X; Z) = I(X; Z_{\text{imp}}) + I(X; Z_{\text{unimp}} \mid Z_{\text{imp}})$, Lemma 1 bounds the label-relevant leakage through unimportant nodes by the second term, so reducing it does not cost prediction. Empirically (Appx. AB) the noise component contributes 5-15% in Fid$\pm$ and a 30-40% reduction in epochs to convergence, while the MLP provides much of the accuracy gain.

Selective Forgetting of Information Our approach compresses information by selectively forgetting about unimportant nodes ($I(X; Z) \downarrow$) while preserving structure around important ones. Gaussianization is the natural mechanism for information loss. By the Central Limit Theorem, repeated independent noise-injection drives convergence towards Gaussianity (the embeddings aggregate many independent perturbations). We inject noise via Equation 3 at each unimportant node.

Below we formalize the theoretical groundings of BottleneckMLP.

Theorem 1 (Mutual information reduction)**.** *Let $Z_{\text{unimp}} = (Z_i)_{i \in \mathcal{U}}$, $Z_{-i} = (Z_j)_{j \in \mathcal{U}, j \neq i}$ and $\tilde{Z}_{\text{unimp}} = \psi(Z_{\text{unimp}})$, then:*

1. *$I(X; Z_{\text{unimp}})$ is non-increasing in each t_i, with $\frac{\partial}{\partial t_i} I(X; Z_{\text{unimp}}) = -\frac{1}{2t_i^2} \operatorname{mmse}\big(1/t_i \,\big|\, Z_{-i}\big) \leq 0$, where $\operatorname{mmse}(\gamma \mid Z_{-i}) = \mathbb{E}\big\| f_i(X) - \mathbb{E}[f_i(X) \mid \sqrt{\gamma} f_i(X) + \epsilon_i,\, Z_{-i}] \big\|^2$. The decrease is strict whenever $f_i(X)$ is not almost surely determined by Z_{-i}, equivalently whenever $I(X; Z_i \mid Z_{-i}) > 0$.*
2. *$I(X; \tilde{Z}_{\text{unimp}}) \leq I(X; Z_{\text{unimp}}) \leq I(X; f(X))$. Equality is attainable for injective ψ; no strict decrease is claimed for the deterministic step.*
3. *$\lim_{\min_i t_i \to \infty} I(X; \tilde{Z}_{\text{unimp}}) = 0$.*

Lemma 1 (Predictive leakage is bounded by compression)**.** *$Y \to X \to Z$ is Markov and $I(Z_{\text{unimp}}; Y \mid Z_{\text{imp}}) \leq I(Z_{\text{unimp}}; X \mid Z_{\text{imp}})$.*

As such, $I(Z_{\text{unimp}}; Y \mid Z_{\text{imp}}) = 0$ is neither assumed nor inferred from held-out accuracy, but rather, the label-relevant leakage through unimportant nodes vanishes at least as fast as the compression term the module drives down. Appx. F evaluates both sides, the left side with a lower bound probe, and the right with a closed form bound of Theorem 1. Across all eight model/dataset pairs the inequality holds with at least two orders of magnitude of margin. In six the true pairing is indistinguishable from a permutation null, so no residual leakage is detectable; in the remaining two it is detectable and amounts to under 1% of the corresponding bound. We therefore state the conditional independence as approximate and bounded rather than exact.

Proposition 2 (Importance-graded information retention)**.** *Let $C_i = \operatorname{Cov}(f_i(X))$ have eigenvalues $\lambda_j^{(i)}$ and let $N_i := \frac{1}{2\pi e} \exp\big(\frac{2}{d} h(f_i(X))\big)$ be the entropy power of the signal. Then for every node i,*

$$\frac{d}{2} \log\Big(1 + \frac{\alpha_i N_i}{\sigma}\Big) \;\leq\; I(X; Z_i) \;\leq\; \frac{1}{2} \sum_{j=1}^{d} \log\Big(1 + \frac{\alpha_i \lambda_j^{(i)}}{\sigma}\Big).$$

For an important node p and an unimportant node q with comparable signal spectra, the retained-information gap is therefore controlled by the importance ratio α_p / α_q and degrades gracefully in σ, with no limit $\sigma \to \infty$ required.

Proposition 2 establishes the separation between important and unimportant nodes that the representation-dynamics experiments of Section 5.1 measure. Since $G_S = g_\phi(\tilde{Z})$ is a function of the module output, $X \to \tilde{Z} \to G_S$ is Markov and $I(X; G_S) \leq I(X; \tilde{Z})$. Every upper bound in Theorem 1 and Proposition 2 therefore transfers to the compression term of the IB objective optimized by the explainer. The DPI is one sided, so the lower bound does not transfer; it constrains

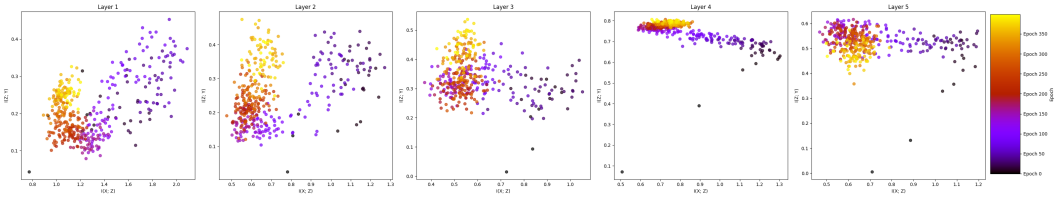


Figure 3: BottleneckMLP enforces IB on graph embeddings in the information plane $I(X;Z_i)$ vs. $I(Z_i;Y)$ over layers, i in range 1–5, and epochs (purple to yellow). Information curve (Tishby & Zaslavsky, 2015) naturally appears with CE minimization. Compression and concentration are more apparent in the later layers, consistent with the theory in Section 4.2.1. X-axis in range (0, 2) for i=1, and (0, 1.3) for i=2–5. Y-axis in range (0,0.5) for i=1–3, (0, 0.8) for i=4, and (0, 0.6) for i=5.

the information retained in Z_i, not what the extractor forwards into G_S. Only the upper direction is needed here, since every claim about compression is an upper bound and the separation is both stated and measured at the module output. These results establish compression, bounded leakage, and importance-graded retention. Faithfulness of the resulting explanations is supported by Tables 3 and 4 and the visualizations in Appx. H; the quality of the importance scores is examined in Appx. Z.

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.

4.2.2 BottleneckMLP Effectively Enforces IB Dynamics

In Figure 1c, $I(X;Z)$ 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 (Appx. I.4). By contrast, explicit IB losses fail: in 1a (GSAT) and 1b (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 architecture-driven approach. The resultant improvement in explainer performance is presented in Section 5.

BottleneckMLP reproduces the information plane dynamics originally observed in DNNs by (Tishby & Zaslavsky, 2015), now extended to the graph domain with GNNs (Figure 3). Both the fitting and compression phases are clearly visible. This two-phase IB trajectory does not appear in the baseline GNN training with or without Info Loss; we include these plots, respectively, for comparison to Figure 3 in Appx. U. In Figure 3, early layers behave similarly to those in the baseline model, but beginning around Layer 3, clear compression in $(I(X;Z_i))$ emerges as training progresses. Points transition from right (high $(I(X;Z_i))$) to left (compressed representations) while $(I(Z_i;Y))$ becomes more concentrated. By Layers 4 and 5, the IB trajectory is visible. This trend aligns with theoretical expectations and with our discussion in Section 4.2.1, demonstrating that BottleneckMLP successfully induces the expected IB behavior in graph representations.

5 Experiments

Runtime analysis is in Appx. S, and the component ablation of the scaled-noise term is in Appx. AB.

5.1 Representation Dynamics of BottleneckMLP Reflect IB

BottleneckMLP induces representation dynamics aligned with the IB framework, selectively reducing $I(X;Z)$ while preserving $I(Z;Y)$: important node embeddings maintain lower entropy and more structured distributions, while unimportant embeddings approach Gaussian-like high-entropy distributions. We summarize LNSA and drift results below, and defer convex hull volume (Appx. I.2), node linkage (Appx. I.3) and HSIC (Appx. I.4) to the appendix.

Table 2: PGIB / GSAT Classifier Accuracy. BottleneckMLP consistently increases performance over the original method and when IB Loss is removed (with the exception of PGIB–NCI1).

Method	MUTAG	BA-2Motifs	NCI1	PROTEINS	Benzene	Alk. Carbonyl	Flu. Carbonyl	IMDB-B	molhiv
GSAT	0.909±.033	0.994±.006	0.689±.017	0.681±.036	1.000	0.907±.151	0.809±.005	0.698±.028	0.765±.019
GSAT w/o IB Loss	0.935±.043	0.770±.224	0.743±.024	0.706±.040	1.000	0.939±.026	0.958±.077	0.694±.038	0.764±.011
GSAT w/o IB + **Ours**	**0.949**±.010	**1.000**	**0.802**±.019	**0.745**±.048	**1.000**	**0.995**±.006	**0.987**±.013	**0.716**±.038	**0.770**±.023
PGIB	0.904±.010	0.628±.171	0.729±.024	0.729±.024	0.886±.008	0.994±.002	0.989±.011	0.728±.008	0.755±.003
PGIB w/o IB Loss	0.916±.011	0.896±.145	**0.774**±.014	0.763±.022	0.899±.006	0.987±.013	0.939±.002	0.710±.026	0.722±.007
PGIB w/o IB + **Ours**	**0.925**±.009	**0.963**±.016	0.753±.019	**0.792**±.018	**0.900**±.004	**0.995**±.000	**0.993**±.003	**0.750**±.010	**0.770**±.006

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) (figure and details in Appx. P). 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 12 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.

Results on Embedding Drift Figure 7 (Appx. I.1) 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 our theory: important nodes stabilize with low drift, while unimportant nodes continue drifting toward noisier, Gaussian-like distributions. This pattern holds across all datasets (see Appx. I.1).

5.2 BottleneckMLP Improves IB-Based Explainers

Table 3: BottleneckMLP consistently improves Fid+/-. (AC/ FC: Alkane-/Fluorine- Carbonyl

Dataset	PGIB		PGIB w/o IB		PGIB + Ours	
	$Fid^+ \uparrow$	$Fid^- \downarrow$	$Fid^+ \uparrow$	$Fid^- \downarrow$	$Fid^+ \uparrow$	$Fid^- \downarrow$
MUTAG	0.75±.08	0.59±.20	0.72±.08	0.52±.19	**0.76**±.07	**0.38**±.25
BA-2M	0.83±.16	0.49±.15	0.83±.16	0.48±.14	**0.98**±.08	**0.40**±.24
NCI1	0.45±.12	0.52±.16	0.52±.14	0.55±.13	**0.77**±.16	**0.48**±.21
PROTEINS	0.64±.02	0.60±.13	0.65±.03	0.60±.04	**0.66**±.02	**0.59**±.09
Benzene	0.51±.02	0.04±.00	0.52±.02	0.04±.01	**0.64**±.05	**0.04**±.00
AC	0.58±.04	0.18±.08	0.92±.04	0.10±.05	**0.97**±.02	**0.09**±.03
FC	0.77±.03	0.15±.05	0.79±.07	0.14±.05	**0.92**±.03	**0.08**±.03

Table 2 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, showing that BottleneckMLP is a more effective realization of IB Loss than the variational bound. AUC-ROC scores and information planes for GSAT variants are in Appx. K. Visual comparisons of explanatory subgraphs are in Appx. H. Table 3 reports fidelity metrics (Appx. O) for PGIB. BottleneckMLP consistently improves Fid^+ and reduces Fid^-, giving higher-quality explanations for all 7 datasets. Table 4 reports improved Average Precision (main metric) and AUC-ROC for TGIB with BottleneckMLP on all 3 datasets. Classifier accuracy is not reported on the original TGIB paper; we include it in the Appx. M.

Table 4: 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 + BottleneckMLP	**0.820**	**0.843**	**0.994**	**0.632**	**0.703**	**0.989**

5.3 BottleneckMLP Generalizes Across Explainability Methods

We use all existing ante-hoc graph explainers with variational IB objectives as baselines, to show directly that our method replaces the explicit IB term. To test generality we additionally apply BottleneckMLP to subgraph recognition (Yu et al., 2022), post-hoc node classification (Luo et al., 2020b), non-IB-based ante-hoc explainers (Wu et al., 2022; Lu et al., 2024), and non-IB-based post-hoc explainers (Wang et al., 2023; 2021). Table 5 audits every setting. BottleneckMLP improves performance in five of eight, recovers the ablated regularizer without strictly exceeding the baseline in two, and is inconclusive in one where all variants sit near chance. Exceptions are individually small and are listed in the table rather than aggregated away. Full per-metric results are in Appx. M, and the comparison with dropout is in Appx. AA.

Table 5: Audit of BottleneckMLP across explainer types and tasks. "Recovers" means performance lost by ablating the explainer's IB loss is at least restored. Percentages are relative to the unmodified baseline. Exceptions are listed in full; per-metric values are in Appx. M.

Explainer	Setting	Verdict	Effect and exceptions	Tab.
VGIB	ante-hoc, subgraph rec.	Improves	Accuracy $+3.5$–62.5%	23
PGExp.	post-hoc, node cls.	Improves	Accuracy $+1.1$–4.9%; expl. AUC-ROC $+0.6$–0.8% on BA-Shapes and TreeGrids, -0.3% on TreeCycles. Recovers the entropy loss, whose removal collapses AUC-ROC to 0.033–0.628	25
GNNExp.	post-hoc, node cls.	Inconcl.	All variants within 0.493–0.510 AUC on all four datasets; the setting does not discriminate	26
V-InFoR	post-hoc, IB-based	Improves	$+1.3\%$ to $+39.4\%$ across metrics. Decreases on BA3 Prob-N (-0.0003) and MUTAG Mean Probability	27
ReFine	post-hoc, graph cls.	Recovers	ACC-AUC $+19.9\%$ on MUTAG; -1.3% on BA3 and $+0.8\%$ on MNIST, fully recovering the fidelity loss	28
ProxyExp.	post-hoc, graph cls.	Mixed	AUC $+31\%$ on Fluoride Carbonyl and $+150\%$ on Alkane Carbonyl; Fid^+ $0.112 \rightarrow 0.775$ on Alkane Carbonyl. Decreases on BA-2Motifs, MUTAG, and Fluoride Carbonyl	29
GOAt	ante-hoc, non-IB	Improves	Fid^+ $+5.2\%$ to $+70.5\%$; sparsity improves in all nine backbone/dataset pairs. Minor decreases on BA-2Motifs for GCN and GIN	30
DIR	ante-hoc, non-IB	Mixed	Train accuracy $+8.4$–41.4%, validation $+0.2$–35.9%, confounding accuracy -0.7 to -8.0%; causal accuracy and test MRR remain below the original on all splits. Recovers ablated Info Loss	31

The hyperparameter σ in Equation 3 controls the node-embedding perturbation strength and is tuned on validation sets with grid-search. A full sensitivity analysis is provided in Appx. X. We further observe that BottleneckMLP accelerates convergence by inducing the desired representation dynamics throughout training. Empirical results for this speed-up are reported in Table 36, Appx. T.

6 Discussion, Limitations, and Future Directions

We present a representation-centric theory of explainability in GNNs, grounded in the IB principle. We show that explanations emerge from representation dynamics during training. BottleneckMLP implements this idea as a model-agnostic architectural module that implicitly enforces an information bottleneck, without explicit loss terms, via importance-weighted Gaussianization and progressive compression through MLP layers. We theoretically and empirically show that these mechanisms induce selective forgetting, and yield interpretable representations, outperforming explicit constraints. Explainability thus arises not as an auxiliary objective, but as a natural consequence of learning what to forget. Formalizing the conditions under which implicit bottlenecks arise and identifying optimal BottleneckMLP architectures for specific datasets and explainers remain open limitations. A deeper information-theoretic analysis of layer-wise compression under graph message passing could help unify our observations with general learning theory.

AI USE STATEMENT

We used generative AI tools minimally to help with basic tasks which do not involve intellectual contributions. We wrote all original text ourselves and occasionally used AI to edit prose for concision and clarity. We also used AI to help with the implementation of some supporting analysis and figure code. We did not use AI for any implementation of our main method. All experiments were designed and run by the authors, and their outputs were inspected by the authors. All numerical claims were checked against the saved outputs. The authors take full responsibility for the content, claims, code, figures, and artifacts of this work.

ETHICS STATEMENT

We believe that the ethical and societal implications of this work are consistent with those commonly encountered in machine learning research for scientific discovery, and we do not identify any specific risks that warrant additional discussion beyond standard considerations (i.e. responsible use).

REPRODUCIBILITY STATEMENT

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

REFERENCES

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation, 2017. URL https://arxiv.org/abs/1611.01353.

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

Alexander A. Alemi, Ian Fischer, and Joshua V. Dillon. Uncertainty in the variational information bottleneck, 2018. URL https://arxiv.org/abs/1807.00906.

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., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/2c8c3a57383c63caef6724343eb62257-Paper.pdf.

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.

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.

Chi-Ning Chou, Hang Le, Yichen Wang, and SueYeon Chung. Feature learning beyond the lazy-rich dichotomy: Insights from representational geometry, 2025. URL https://arxiv.org/abs/2503.18114.

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.

Dongning Guo, S. Shamai, and S. Verdu. Mutual information and minimum mean-square error in gaussian channels. *IEEE Transactions on Information Theory*, 51(4):1261–1282, 2005. doi: 10.1109/TIT.2005.844072.

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.

Shengyao Lu, Keith G. Mills, Jiao He, Bang Liu, and Di Niu. Goat: Explaining graph neural networks via graph output attribution, 2024. URL https://arxiv.org/abs/2401.14578.

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.

Morteza Noshad, Yu Zeng, and Alfred O. Hero III. Scalable mutual information estimation using dependence graphs, 2018. URL https://arxiv.org/abs/1801.09125.

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.

Senzhang Wang, Jun Yin, Chaozhuo Li, Xing Xie, and Jianxin Wang. V-infor: A robust graph neural networks explainer for structurally corrupted graphs. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 56469–56487. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/b07d36fb3fae0630897700593c8cf49d-Paper-Conference.pdf.

Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards multi-grained explainability for 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. 18446–18458. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/99bcfcd754a98ce89cb86f73acc04645-Paper.pdf.

Ying-Xin Wu, Xiang Wang, An Zhang, Xiangnan He, and Tat-Seng Chua. Discovering invariant rationales for graph neural networks, 2022. URL https://arxiv.org/abs/2201.12872.

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 15
B Lemma: Aggregate-posterior identity 15
C Proof of Theorem 1 16
D Explainer-Specific Corollaries 18
E Proposition 4 and Proof 20
F Verification of Theoretical Results 21
G Experimental Setup 24
H Explanatory Subgraph Visualization 25
I Representation Dynamics Across Layers 25
I.1 Node Drift 25
I.2 Convex Hull 26
I.3 Node Linkage Distance 26
I.4 HSIC Results 28
J Link Prediction 29
K AUC-ROC 29
L Testing BottleneckMLP on a Variety of Architectures 30
M Generalization of BottleneckMLP across Graph Tasks 30
N Explicit IB Methods 33
O Fidelity Metric 35
P Normalized Space Alignment (NSA) 35
Q Derivations 35
R PGIB 36
S Running Time Analysis 36

T Convergence Analysis . . . 36
U Information Plane Plots for Baseline Models . . . 37
V Pseudocode . . . 38
W GNN Backbones . . . 39
X Hyperparameter Sensitivity Analysis . . . 39
Y Sparsity . . . 39
Z Robustness to Importance Score Quality . . . 41
AA Comparison with Dropout and Noise-Injection Baselines . . . 42
AB BottleneckMLP Component Ablation . . . 43
AC EDGE Validation as MI Estimator . . . 44

A List of Symbols

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
$\psi(\cdot)$	Deterministic mapping function (MLP)
$\tilde{Z}_{unimp}$	Deterministic mapping of Z_{unimp}) by ψ
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
$J(\cdot)$	Fisher information matrix
$Tr(\cdot)$	Trace of a matrix
det	Matrix determinant
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
ΔI_X^k	Partial compression at layer k
h_v^k	Embedding of node v in kth layer
'IB Loss'	Loss term that explicitly enforces IB principle

B Lemma: Aggregate-posterior identity

Lemma 2 (Aggregate-posterior identity)**.** *Let U be a random variable, $P(\cdot \mid U)$ a conditional distribution on a discrete space $\mathcal{Z}$, and Q a fixed distribution on $\mathcal{Z}$ that does not depend on U. Let*

$Z \sim P(\cdot \mid U)$ and write $\bar{P}(\cdot) := \mathbb{E}_U[P(\cdot \mid U)]$ for the aggregate posterior. If $\mathrm{supp}(\bar{P}) \subseteq \mathrm{supp}(Q)$, then

$$\mathbb{E}_U\big[\mathrm{KL}\big(P(\cdot \mid U) \,\big\|\, Q\big)\big] \;=\; I(U;Z) \;+\; \mathrm{KL}\big(\bar{P} \,\|\, Q\big).$$

Both terms are non-negative, so the objective upper-bounds $I(U;Z)$ with the true value gap $\mathrm{KL}(\bar{P}\|Q)$, depending on the prior only through Q and on the encoder only through its average output distribution.

C Proofs

C.1 Proof of Lemma 2

Proof. Let (U, Z) have joint law $P(U)P(Z \mid U)$, so that Z has marginal $\bar{P}$. Since $\mathrm{supp}(\bar{P}) \subseteq \mathrm{supp}(Q)$ all logarithms below are finite. Write

$$\mathbb{E}_U\big[\mathrm{KL}(P(\cdot \mid U)\|Q)\big] \;=\; \mathbb{E}_{U,Z}\left[\log \frac{P(Z \mid U)}{Q(Z)}\right] \;=\; \mathbb{E}_{U,Z}\left[\log \frac{P(Z \mid U)}{\bar{P}(Z)}\right] + \mathbb{E}_{U,Z}\left[\log \frac{\bar{P}(Z)}{Q(Z)}\right].$$

The first term is $I(U;Z)$ by definition, since $\bar{P}$ is the marginal of Z. In the second term the integrand depends on Z alone, and $Z \sim \bar{P}$ marginally, so it equals $\mathrm{KL}(\bar{P}\|Q)$. Both terms are non-negative. □

C.2 Proof of Proposition 1

Proof. **Equation equation 1.** Both $P_\phi(\cdot \mid U)$ and Q are product measures over $\mathcal{K}$, and the KL divergence between product measures is the sum of the coordinatewise divergences:

$$\mathrm{KL}\big(P_\phi(\cdot \mid U) \,\|\, Q\big) = \sum_{k \in \mathcal{K}} \mathrm{KL}\big(\mathrm{Bern}(\phi_k(U)) \,\|\, \mathrm{Bern}(r)\big).$$

Taking $\mathbb{E}_U$ and applying Lemma 2 to each coordinate separately — with $Z = \hat{s}_k$, $P(\cdot \mid U) = \mathrm{Bern}(\phi_k(U))$, $Q_k = \mathrm{Bern}(r)$, and aggregate posterior $\bar{P}_k = \mathrm{Bern}(\bar{\phi}_k)$ where $\bar{\phi}_k = \mathbb{E}_U[\phi_k(U)]$ — gives

$$\mathcal{L}_{\mathrm{IB}} = \sum_{k \in \mathcal{K}} \Big[I(U;\hat{s}_k) + \mathrm{KL}\big(\mathrm{Bern}(\bar{\phi}_k)\|\mathrm{Bern}(r)\big)\Big].$$

Finiteness of each term follows from $\phi_k(U) \in (0,1)$ and $r \in (0,1)$.

Equation equation 2. It suffices to show $\sum_k I(U;\hat{s}_k) - I(U;\hat{s}) = \mathrm{TC}(\bar{P}_\phi)$. By conditional independence of the units given U,

$$H(\hat{s} \mid U) = \sum_{k \in \mathcal{K}} H(\hat{s}_k \mid U).$$

Hence

$$\sum_k I(U;\hat{s}_k) - I(U;\hat{s}) = \sum_k \big[H(\hat{s}_k) - H(\hat{s}_k \mid U)\big] - \big[H(\hat{s}) - H(\hat{s} \mid U)\big] = \sum_k H(\hat{s}_k) - H(\hat{s}),$$

where the conditional entropies cancel. The right-hand side is by definition the total correlation $\mathrm{TC}(\bar{P}_\phi)$ of $\hat{s}$ under its marginal law $\bar{P}_\phi$, and $\mathrm{TC} \geq 0$ by subadditivity of entropy, with equality iff the units are mutually independent under $\bar{P}_\phi$. Substituting into equation 1 gives equation 2. □

Remark 1. *Equivalently, equation 2 follows from Lemma 2 applied to the joint mask together with the standard identity $\mathrm{KL}(\bar{P}_\phi\|Q) = \mathrm{TC}(\bar{P}_\phi) + \sum_k \mathrm{KL}(\bar{P}_k\|Q_k)$ for factorized Q, obtained by inserting and cancelling $\prod_k \bar{P}_k$.*

C.3 Proof of Theorem 1

Proof. **Claim 1 (monotonicity and strictness).** Fix $i \in \mathcal{U}$. Since Z_i depends on X only through $S_i = f_i(X)$ and the noise is independent of X,

$$I(X; Z_i \mid Z_{-i}) = h(Z_i \mid Z_{-i}) - h(Z_i \mid X, Z_{-i}) = h(Z_i \mid Z_{-i}) - h(\sqrt{t_i}\,\epsilon_i),$$

so $I(X; Z_i \mid Z_{-i}) = I(S_i; Z_i \mid Z_{-i})$. By the chain rule,

$$I(X; Z_{\text{unimp}}) = I(X; Z_{-i}) + I(X; Z_i \mid Z_{-i}), \tag{4}$$

and only the second term depends on t_i, because Z_{-i} neither contains ϵ_i nor depends on t_i.

Reparameterize with $\gamma := 1/t_i$. Since scaling by a positive constant is invertible and hence preserves mutual information, $I(X; Z_i \mid Z_{-i}) = I\big(X; \sqrt{\gamma}\, S_i + \epsilon_i \;\big|\; Z_{-i}\big)$. Conditionally on $Z_{-i} = z$, the channel $S_i \mapsto \sqrt{\gamma} S_i + \epsilon_i$ is an additive white Gaussian noise channel whose noise is independent of (S_i, Z_{-i}), so the I-MMSE relation (Guo et al., 2005) applies conditionally:

$$\frac{\partial}{\partial \gamma}\, I\big(X; \sqrt{\gamma} S_i + \epsilon_i \mid Z_{-i} = z\big) = \tfrac{1}{2}\, \text{mmse}(\gamma \mid Z_{-i} = z),$$

with $\text{mmse}(\gamma \mid Z_{-i} = z) = \mathbb{E}\big[\|S_i - \mathbb{E}[S_i \mid \sqrt{\gamma} S_i + \epsilon_i, Z_{-i} = z]\|^2 \,\big|\, Z_{-i} = z\big]$. The condition $\mathbb{E}\|S_i\|^2 < \infty$ licenses differentiation under the expectation over z, giving

$$\frac{\partial}{\partial \gamma} I(X; Z_i \mid Z_{-i}) = \tfrac{1}{2}\, \text{mmse}(\gamma \mid Z_{-i}).$$

Composing with $\partial\gamma/\partial t_i = -1/t_i^2$ and using equation 4,

$$\frac{\partial}{\partial t_i} I(X; Z_{\text{unimp}}) = -\frac{1}{2t_i^2}\, \text{mmse}\big(1/t_i \mid Z_{-i}\big) \;\le\; 0.$$

The MMSE is non-negative, and vanishes at finite γ if and only if S_i is recoverable with zero error from Z_{-i} alone, i.e. S_i is almost surely $\sigma(Z_{-i})$-measurable. In that case Z_i contributes only independent noise beyond Z_{-i} and $I(X; Z_i \mid Z_{-i}) = 0$; conversely if S_i is not a.s. determined by Z_{-i} then $\text{mmse} > 0$ and $I(X; Z_i \mid Z_{-i}) > 0$. The decrease is therefore strict exactly on $\{I(X; Z_i \mid Z_{-i}) > 0\}$.

We note that this argument needs only $\mathbb{E}\|S_i\|^2 < \infty$; no Fisher-information condition is required.

Claim 2 (deterministic step). Since $\tilde{Z}_{\text{unimp}} = \psi(Z_{\text{unimp}})$ with ψ deterministic, $X \to Z_{\text{unimp}} \to \tilde{Z}_{\text{unimp}}$ is Markov and the data processing inequality gives $I(X; \tilde{Z}_{\text{unimp}}) \le I(X; Z_{\text{unimp}})$. Because the noise is injected after the encoder, $X \to f(X) \to Z_{\text{unimp}}$ is Markov and a second application gives $I(X; Z_{\text{unimp}}) \le I(X; f(X))$. Neither inequality is strict in general: if ψ is injective then Z_{unimp} is a deterministic function of $\tilde{Z}_{\text{unimp}}$ as well, so the DPI applies in both directions and equality holds.

Claim 3 (asymptotic limit). By Claim 2 it suffices to show $I(X; Z_{\text{unimp}}) \to 0$. The noise variables are mutually independent, so $I(X; Z_{\text{unimp}}) \le \sum_{i \in \mathcal{U}} I(X; Z_i)$, and it suffices that $I(X; Z_i) \to 0$ for each i. As established above $I(X; Z_i) = I(S_i; S_i + \sqrt{t_i}\epsilon_i)$, and among all signals with covariance C_i the Gaussian maximizes entropy, so

$$I(X; Z_i) \;\le\; \tfrac{1}{2} \log\det\big(I_d + C_i/t_i\big) = \tfrac{1}{2} \sum_{j=1}^{d} \log\Big(1 + \frac{\lambda_j^{(i)}}{t_i}\Big),$$

with $\lambda_j^{(i)}$ finite by $\mathbb{E}\|S_i\|^2 < \infty$. Each summand vanishes as $t_i \to \infty$, giving the claim. □

C.4 Proof of Lemma 1

Proof. The module's noise is independent of (X, Y) and Z is a measurable function of X and that noise, so Z is conditionally independent of Y given X; that is, $Y \to X \to Z$ is Markov, and likewise conditionally on any sub-vector of Z. Hence

$$I(Z_{\text{unimp}}; Y \mid X, Z_{\text{imp}}) = 0.$$

By the chain rule for mutual information,

$$I(Z_{\text{unimp}}; X, Y \mid Z_{\text{imp}}) = I(Z_{\text{unimp}}; X \mid Z_{\text{imp}}) + I(Z_{\text{unimp}}; Y \mid X, Z_{\text{imp}}) = I(Z_{\text{unimp}}; X \mid Z_{\text{imp}}).$$

Monotonicity of mutual information under enlarging the second argument gives $I(Z_{\text{unimp}}; Y \mid Z_{\text{imp}}) \le I(Z_{\text{unimp}}; X, Y \mid Z_{\text{imp}})$, and combining the two displays yields the claim. □

C.5 Proof of Proposition 2

Proof. Throughout, $S_i = f_i(X)$ and, as shown in the proof of Theorem 1, $I(X; Z_i) = I(S_i; S_i + \sqrt{t_i}\epsilon_i) = h(Z_i) - h(\sqrt{t_i}\epsilon_i)$, with $h(\sqrt{t_i}\epsilon_i) = \frac{d}{2}\log(2\pi e\, t_i)$.

Upper bound. $\mathrm{Cov}(Z_i) = C_i + t_i I_d$, and the Gaussian maximizes differential entropy for a given covariance, so $h(Z_i) \le \frac{1}{2}\log\big((2\pi e)^d \det(C_i + t_i I_d)\big)$. Therefore

$$I(X; Z_i) \le \tfrac{1}{2}\log\frac{\det(C_i + t_i I_d)}{t_i^d} = \tfrac{1}{2}\sum_{j=1}^{d}\log\Big(1 + \frac{\lambda_j^{(i)}}{t_i}\Big).$$

Lower bound. S_i and $\sqrt{t_i}\epsilon_i$ are independent and both admit densities, so the entropy power inequality applies to $Z_i = S_i + \sqrt{t_i}\epsilon_i$:

$$N_e(Z_i) \;\ge\; N_e(S_i) + N_e\big(\sqrt{t_i}\epsilon_i\big) = N_i + t_i,$$

using $N_e(\sqrt{t_i}\epsilon_i) = t_i$ for isotropic Gaussian noise. Since $h(Z_i) = \frac{d}{2}\log\big(2\pi e\, N_e(Z_i)\big)$ by definition of entropy power,

$$I(X; Z_i) = h(Z_i) - \tfrac{d}{2}\log(2\pi e\, t_i) = \tfrac{d}{2}\log\frac{N_e(Z_i)}{t_i} \;\ge\; \tfrac{d}{2}\log\Big(1 + \frac{N_i}{t_i}\Big).$$

Substituting $t_i = \sigma/\alpha_i$ into both bounds gives the stated display. The final sentence follows by comparing the bounds at α_p and α_q: for nodes with comparable N_i and spectra, the lower bound at p and the upper bound at q are separated whenever α_p/α_q is large relative to σ scaled by the signal energy, and both bounds are finite and continuous in σ on $(0, \infty)$. □

D Explainer-Specific Corollaries

Proposition 1 introduces the gap that exists with the use of variational upper bounds on MI. In this section, we give explainer-specififc corollaries showing why the specific variational upper bounds derived in each of the existing IB explainer methods do not suffice.

Corollary 1 (GSAT, with variable graph size)**.** *For GSAT, $U = G$, the units are the edges $\mathcal{K} = E(G)$, and $\hat{e}_{ij} \sim \mathrm{Bern}(\phi_{ij}(G))$ against the prior $\mathrm{Bern}(r)$, so Proposition 1 applies verbatim whenever all graphs share a common edge set.*

For a corpus of graphs of varying size, let $m(G) := |E(G)|$ and $\bar{m} := \mathbb{E}_P[m(G)] < \infty$, and define the size-biased law $\tilde{P}$ *by*

$$\tilde{P}(G) \;:=\; \frac{m(G)\,P(G)}{\bar{m}}, \qquad \tilde{P}(E \mid G) \;:=\; \frac{1}{m(G)} \;\; \textit{for } E \in E(G),$$

i.e. draw an edge uniformly from the pooled edge population of the corpus and let G be the graph it came from. Write $\bar{\phi} := \mathbb{E}_{\tilde{P}}[\phi_E(G)]$ for the pooled mean attention. Then

$$\mathcal{L}_{VUB} \;=\; \bar{m}\Big[\underbrace{I_{\tilde{P}}\big((G, E); \hat{e}_E\big)}_{\textit{per-edge information}} + \underbrace{\mathrm{KL}\big(\mathrm{Bern}(\bar{\phi}) \,\|\, \mathrm{Bern}(r)\big)}_{\textit{marginal mismatch}}\Big],$$

with both terms non-negative. The decomposition is exact and requires no assumption on graph size or on a correspondence between edges of different graphs.

Proof. Write $F(p) := \mathrm{KL}(\mathrm{Bern}(p)\|\mathrm{Bern}(r))$, so that by factorization of the KL between product measures, $\mathcal{L}_{VUB} = \mathbb{E}_P\big[\sum_{ij\in E(G)} F(\phi_{ij}(G))\big]$.

Step 1: size-biasing. For any measurable f,

$$\mathbb{E}_{\tilde{P}}\big[f(\phi_E(G))\big] = \sum_G \frac{m(G)P(G)}{\bar{m}} \cdot \frac{1}{m(G)} \sum_{ij\in E(G)} f(\phi_{ij}(G)) = \frac{1}{\bar{m}}\,\mathbb{E}_P\Big[\sum_{ij\in E(G)} f(\phi_{ij}(G))\Big].$$

Taking $f = F$ and multiplying by $\bar{m}$ gives $\mathcal{L}_{VUB} = \bar{m}\,\mathbb{E}_{\tilde{P}}\big[\mathrm{KL}(\mathrm{Bern}(\phi_E(G))\|\mathrm{Bern}(r))\big]$. Note that $\tilde{P}$ is a probability measure precisely because $\sum_G m(G)P(G) = \bar{m}$, and that $\bar{m} < \infty$ is needed for this normalization.

Step 2: the aggregate-posterior identity. Apply Lemma 2 under $\tilde{P}$ with conditioning variable $U = (G, E)$, observation $Z = \hat{e}_E \sim \mathrm{Bern}(\phi_E(G))$, and fixed prior $Q = \mathrm{Bern}(r)$. The aggregate posterior is $\bar{P} = \mathrm{Bern}(\mathbb{E}_{\tilde{P}}[\phi_E(G)]) = \mathrm{Bern}(\bar{\phi})$, so

$$\mathbb{E}_{\tilde{P}}\big[\mathrm{KL}(\mathrm{Bern}(\phi_E(G))\|\mathrm{Bern}(r))\big] = I_{\tilde{P}}\big((G,E);\hat{e}_E\big) + \mathrm{KL}\big(\mathrm{Bern}(\bar{\phi})\|\mathrm{Bern}(r)\big).$$

Multiplying by $\bar{m}$ gives the claim. Both terms are non-negative, and both are finite since $\phi_{ij}(G) \in (0,1)$ and $r \in (0,1)$. □

The first term is a per-edge quantity under $\tilde{P}$, not the joint target $I(G; G_S)$; relating the two requires the redundancy term of Proposition 1, which is defined only on a fixed unit set.

Corollary 2 (Certified lower bound on the gap, variable graph size)**.** *Let $M := m(G)$ be the number of edges, with $\bar{m} = \mathbb{E}_P[M] < \infty$ and $H(M)$ the entropy of its distribution under P. Let $\bar{\phi}$ be the size-biased pooled mean attention of Corollary 1. Then*

$$\mathcal{L}_{VUB} - I(G;\hat{s}) \;\geq\; \bar{m}\,\mathrm{KL}\big(\mathrm{Bern}(\bar{\phi})\,\|\,\mathrm{Bern}(r)\big) \;-\; H(M).$$

Both terms are computable in closed form from logged attention and the graph-size histogram; no correspondence between edges of different graphs is required.

Proof. Since $\hat{s}$ is a binary vector of length M, observing it determines M, so $H(\hat{s}) = H(M) + H(\hat{s} \mid M)$. Fix any ordering of the edges within each graph. Subadditivity of entropy holds for any such ordering, so conditioning on the stratum $\{M = k\}$ with weight $n_k := P(M = k)$,

$$H(\hat{s} \mid M) \;\leq\; \sum_k n_k \sum_{j \leq k} H\big(\hat{s}_j \mid M = k\big) \;=\; \sum_k n_k \sum_{j \leq k} H_b\big(\bar{\phi}_{j|k}\big), \qquad \bar{\phi}_{j|k} := \mathbb{E}[\phi_j(G) \mid M = k].$$

Given G the mask is factorized, so $H(\hat{s} \mid G) = \sum_k n_k \sum_{j \leq k} \mathbb{E}[H_b(\phi_j(G)) \mid M = k]$ exactly. Subtracting,

$$I(G;\hat{s}) = H(\hat{s}) - H(\hat{s} \mid G) \;\leq\; H(M) + \sum_k n_k \sum_{j \leq k} I\big(G; \hat{s}_j \mid M = k\big).$$

Within each stratum the unit set is fixed, so Proposition 1 applies and gives $\sum_{j \leq k} I(G; \hat{s}_j \mid M = k) = \mathcal{L}^{(k)}_{VUB} - \sum_{j \leq k} \mathrm{KL}(\mathrm{Bern}(\bar{\phi}_{j|k})\|\mathrm{Bern}(r))$, where $\mathcal{L}^{(k)}_{VUB}$ is the loss restricted to that stratum. Averaging over strata, $\sum_k n_k \mathcal{L}^{(k)}_{VUB} = \mathcal{L}_{VUB}$, hence

$$\mathcal{L}_{VUB} - I(G;\hat{s}) \;\geq\; \sum_k n_k \sum_{j \leq k} \mathrm{KL}\big(\mathrm{Bern}(\bar{\phi}_{j|k})\|\mathrm{Bern}(r)\big) - H(M).$$

Finally, $p \mapsto \mathrm{KL}(\mathrm{Bern}(p)\|\mathrm{Bern}(r))$ is strictly convex, and the weights n_k over the (k, j) pairs have total mass $\sum_k n_k\, k = \bar{m}$ with weighted mean $\bar{\phi}$. Jensen's inequality gives $\sum_k n_k \sum_{j \leq k} \mathrm{KL}(\mathrm{Bern}(\bar{\phi}_{j|k})\|\mathrm{Bern}(r)) \geq \bar{m}\,\mathrm{KL}(\mathrm{Bern}(\bar{\phi})\|\mathrm{Bern}(r))$. □

Corollary 3 (TGIB)**.** *For TGIB, fix a target edge e_k and let $U_k = (e_k, G_k)$ be its query, with $\mathcal{K}_k$ the candidate events in the L-hop temporal neighbourhood and $P_\phi(R_k \mid U_k) = \prod_{u \in \mathcal{K}_k} \mathrm{Bern}(\phi_u)$ against the prior $\mathrm{Bern}(r)$. Then Proposition 1 applies per target edge with $\mathcal{K} = \mathcal{K}_k$.*

Averaging over target edges, let $n(U) := |\mathcal{K}_U|$ and $\bar{n} := \mathbb{E}[n(U)] < \infty$, and define the size-biased law $\tilde{P}(U) := n(U)P(U)/\bar{n}$ with $V \mid U$ uniform on $\mathcal{K}_U$. Writing $\bar{\phi} := \mathbb{E}_{\tilde{P}}[\phi_V(U)]$,

$$\mathcal{L}^{\mathrm{TGIB}}_{\mathrm{IB}} \;=\; \bar{n}\Big[\, I_{\tilde{P}}\big((U,V);\hat{s}_V\big) \;+\; \mathrm{KL}\big(\mathrm{Bern}(\bar{\phi})\,\|\,\mathrm{Bern}(r)\big)\Big],$$

and, with $H(N)$ the entropy of the neighbourhood-size distribution,

$$\mathcal{L}^{\mathrm{TGIB}}_{\mathrm{IB}} - I(U;\hat{s}) \;\geq\; \bar{n}\,\mathrm{KL}\big(\mathrm{Bern}(\bar{\phi})\,\|\,\mathrm{Bern}(r)\big) \;-\; H(N).$$

Proof. For fixed U_k the candidate set is finite and the posterior over R_k is a product of independent Bernoulli variables, so Proposition 1 applies verbatim with $U = U_k$ and $\mathcal{K} = \mathcal{K}_k$. The two displays then follow from the proofs of Corollaries 1 and 2 with $m(G)$ replaced by $n(U)$ and $\bar{m}$ by $\bar{n}$; nothing in either argument uses the fact that the units are edges of a static graph. □

Remark 2 (TGIB has the largest redundancy term)**.** *Candidate events within one temporal neighborhood share state through the recurrent memory, so consecutive interactions of the same node are strongly dependent about U_k. The redundancy term in Equation 2 therefore absorbs temporal autocorrelation in addition to structural entanglement, predicting the largest gap-to-loss ratio of the three explainers. This is consistent with Table 4, where removing the IB loss improves average precision on CanParl and Wikipedia.*

Corollary 4 (PGIB, variational branch)**.** *When PGIB's subgraph selector uses an edge-independent Bernoulli posterior against a Bernoulli prior, Corollary 1 applies unchanged with $U = G$. We note that the prototype term $I(Y; G_p \mid G_S)$ is a predictive term rather than a compression term and lies outside the scope of Proposition 1.*

Moreover, the contrastive version of PGIB provides no compression, per the reasoning below. Let $\mathcal{L}_{\mathrm{CONT}}$ be the InfoNCE-type objective of Eq. 7 with K contrasted samples and critic g, and let $\hat{I}_{\mathrm{NCE}} := -\mathcal{L}_{\mathrm{CONT}}$. Then for every critic g and every joint distribution,

$$\hat{I}_{\mathrm{NCE}} \;\le\; I(\,\cdot\,;\cdot\,) \qquad \text{and} \qquad \hat{I}_{\mathrm{NCE}} \;\le\; \log K.$$

Due to this, decreasing $\hat{I}_{\mathrm{NCE}}$ places no upper constraint on the mutual information it bounds, which may be unchanged or larger, and any mutual information exceeding $\log K$ is indistinguishable from $\log K$ under this estimator. Hence $\mathcal{L}_{\mathrm{CONT}}$ cannot certify a reduction of $I(G; G_S)$. We therefore need to consider the following proposition.

Proposition 3 (Contrastive surrogates provide no compression certificate)**.** *Let $\mathcal{L}_{\mathrm{CONT}}$ be the InfoNCE-type objective of Equation 7 with K contrasted samples and critic g, and write $\hat{I}_{\mathrm{NCE}} := -\mathcal{L}_{\mathrm{CONT}}$. Then for every critic g and every joint distribution,*

$$\hat{I}_{\mathrm{NCE}} \;\le\; I(\,\cdot\,;\cdot\,) \qquad \textit{and} \qquad \hat{I}_{\mathrm{NCE}} \;\le\; \log K.$$

Consequently: (i) decreasing $\hat{I}_{\mathrm{NCE}}$ places no upper constraint on the mutual information it bounds, which may be unchanged or larger; and (ii) any mutual information exceeding $\log K$ is indistinguishable from $\log K$ under this estimator. Hence $\mathcal{L}_{\mathrm{CONT}}$ cannot certify a reduction of $I(G; G_S)$ at any critic quality, and Proposition 1 does not apply to this branch.

Proof. With $\{y_j\}_{j=1}^{K}$ of which y_1 is drawn jointly with x and the rest marginally,

$$\hat{I}_{\mathrm{NCE}} = \mathbb{E}\left[\log \frac{e^{g(x,y_1)}}{\frac{1}{K}\sum_{j=1}^{K} e^{g(x,y_j)}}\right].$$

That $\hat{I}_{\mathrm{NCE}} \le I(x; y)$ for every g is the standard InfoNCE lower-bound property (Poole et al., 2019). For the ceiling, every term of the denominator is positive, so $\frac{1}{K}\sum_j e^{g(x,y_j)} \ge \frac{1}{K} e^{g(x,y_1)}$ and hence pointwise

$$\log \frac{e^{g(x,y_1)}}{\frac{1}{K}\sum_j e^{g(x,y_j)}} \;\le\; \log \frac{e^{g(x,y_1)}}{\frac{1}{K} e^{g(x,y_1)}} = \log K;$$

taking expectations gives $\hat{I}_{\mathrm{NCE}} \le \log K$.

Claim (i) follows because a lower bound constrains the true quantity only from below: $\hat{I}_{\mathrm{NCE}} \le c$ is consistent with every value $I(x; y) \ge \hat{I}_{\mathrm{NCE}}$. Claim (ii) follows from the ceiling, since all joint distributions with $I(x; y) > \log K$ share the same attainable maximum and are therefore indistinguishable under the estimator. □

E Proposition 4 and Proof

Proposition 4 (Bottleneck capacity)**.** *Let $Z = f(X) + \sqrt{t}\,\epsilon$ with $\epsilon \sim \mathcal{N}(0, I_d)$ independent of X, and let $C = \mathrm{Cov}(f(X))$ have eigenvalues $\lambda_1 \ge \cdots \ge \lambda_d$ with $\mathbb{E}\|f(X)\|^2 < \infty$. Suppose the*

module factors as $\tilde{Z} = \psi_2(W_1 Z)$ for some $W_1 \in \mathbb{R}^{w\times d}$ of full row rank $w \le d$ and an arbitrary deterministic ψ_2. Then

$$I(X;\tilde{Z}) \;\le\; \frac{1}{2}\sum_{j=1}^{w}\log\Big(1+\frac{\lambda_j}{t}\Big).$$

The bound depends on the bottleneck width w and not on the output dimension, and is independent of ψ_2 and of the scale of W_1.

Proof. Write $B = W_1 Z$. Since ψ_2 is deterministic, $X \to Z \to B \to \tilde{Z}$ is Markov and the data processing inequality gives $I(X;\tilde{Z}) \le I(X;B)$.

Now $B = W_1 f(X) + \sqrt{t}\, W_1\epsilon$, where $W_1\epsilon \sim \mathcal{N}(0, W_1 W_1^\top)$ is independent of X and non-degenerate because W_1 has full row rank. Conditional on $X = x$ the term $W_1 f(x)$ is a constant, so

$$h(B \mid X) = h\big(\sqrt{t}\, W_1\epsilon\big) = \tfrac{1}{2}\log\big((2\pi e)^w \det(t\, W_1 W_1^\top)\big).$$

Since $\mathrm{Cov}(B) = W_1 C W_1^\top + t\, W_1 W_1^\top$ and the Gaussian maximizes differential entropy at fixed covariance,

$$I(X;B) = h(B) - h(B \mid X) \;\le\; \tfrac{1}{2}\log\frac{\det\big(W_1 C W_1^\top + t\, W_1 W_1^\top\big)}{\det\big(t\, W_1 W_1^\top\big)} \;=\; \tfrac{1}{2}\log\det\Big(I_w + \tfrac{1}{t}\,(W_1 W_1^\top)^{-1} W_1 C W_1^\top\Big).$$

Take the singular value decomposition $W_1 = U\Sigma V^\top$ with U orthogonal $w\times w$, Σ invertible diagonal, and $V \in \mathbb{R}^{d\times w}$ having orthonormal columns. Then

$$(W_1 W_1^\top)^{-1} W_1 C W_1^\top = U\Sigma^{-1}\big(V^\top C V\big)\Sigma U^\top,$$

which is similar to $V^\top C V$; in particular the scale factor Σ does not affect its spectrum. Let $\mu_1 \ge \cdots \ge \mu_w$ be the eigenvalues of $V^\top C V$. By the Poincaré separation theorem, compressing a symmetric matrix by a matrix with orthonormal columns interlaces its spectrum, so $\mu_j \le \lambda_j$ for every $j \le w$. Therefore

$$I(X;\tilde{Z}) \;\le\; \frac{1}{2}\sum_{j=1}^{w}\log\Big(1+\frac{\mu_j}{t}\Big) \;\le\; \frac{1}{2}\sum_{j=1}^{w}\log\Big(1+\frac{\lambda_j}{t}\Big). \qquad \square$$

Remark 3. *Proposition 4 is stated for a linear compression map, which covers the first layer of the module. For a nonlinear ψ_1 the eigenvalue truncation no longer follows, but the rank statement that motivates it does. Appendix F measures the smooth counterpart of this rank, the participation ratio $(\sum_j \tilde{\lambda}_j)^2/\sum_j \tilde{\lambda}_j^2$, and finds it tracks w while the output dimension is held fixed.*

F Verification of Theoretical Results

In this section we empirically validate all results in Section 4.1.

F.1 Verification of Proposition 1

We implement a factorized-propr explainer that is small enough to exactly enumerate the joint law over masks ($|\mathcal{K}| = 6$ units, 8 equiprobable inputs, and $r = 0.5$ with $\phi_K(U)$ drawn uniformly on $(0.05, 0.95)$. Each of the quantities below is computed by exhaustive enumeration over the 2^6 masks states such that we do not need to use an MI estimator and can directly measure.

The maximum discrepancy for Equation 1 is 0 and for Equation 2 it is 8.3×10^{-16}. In this test, the loss is identical, yet the target differs by 30% and the mismatch term remains unchanged. Therefore, the entire discrepancy is caused by the total correlation term, hence, confirming Proposition 1.

F.2 Verification of Theorem 1, Claim 1

We now verify the monotonicity, the I-MMSE rate, ans the need for the conditional non-redundancy condition in a jointly Gaussian model ($X \sim \mathcal{N}(0, I_p)$, $S = AX$, $Z_i = S_i + \sqrt{t_i}\epsilon_i$) where

Table 6: Exact verification. Rows 2 and 3 use identical per-unit marginals and differ only in which units co-activate on the same input. Identities equation 1 and equation 2 hold to machine precision in all three settings.

	$\mathcal{L}_{\mathrm{IB}}$	$I(U;\hat{s})$	gap	TC	mismatch
generic	0.910798	0.782412	0.128444	0.067997	0.060446
spread	1.096798	0.890059	0.206739	0.120721	0.086018
concentrated	1.096798	0.626681	0.470117	0.384099	0.086018

$I(X;Z_{\mathrm{unimp}})$ has a closed form, so once again no estimator is involved and we have direct measurements.

We sweep t_1 from $[0.3, 3.0]$ with $m = 5$, and $p = 8$. $I(X;Z_{\mathrm{unimp}})$ is non-increasing at every step, and the finite difference derivative matches the $-\mathrm{mmse}(1/t_1 \mid Z_{-1})/(2t_1^2)$ with a worst relative error of 6.5×10^{-10} across the whole sweep.

Table 7 checks that coordinates 0 and 1 have the same signal direction and coordinate 2 has an orthogonal signal direction. We sweep the redundant observer;s noise t_1.

Table 7: The strict-decrease condition must be conditional. The marginal quantity $I(X;Z_0)$ is constant and strictly positive throughout, so the condition $I(X;Z_0) > 0$ holds everywhere; yet the derivative of the joint mutual information collapses by four orders of magnitude as the redundant coordinate becomes noiseless. The conditional quantity $I(X;Z_0 \mid Z_{-0})$ tracks the derivative exactly.

t_1	$I(X;Z_0)$	$I(X;Z_0 \mid Z_{-0})$	$\partial I/\partial t_0$	ratio
10^{-4}	0.21291	5.00×10^{-5}	-5.00×10^{-5}	1.000
10^{-3}	0.21291	4.99×10^{-4}	-4.99×10^{-4}	0.999
10^{-2}	0.21291	4.88×10^{-3}	-4.86×10^{-3}	0.995
10^{-1}	0.21291	4.04×10^{-2}	-3.88×10^{-2}	0.961
10^{0}	0.21291	1.49×10^{-1}	-1.29×10^{-1}	0.865
10^{1}	0.21291	2.04×10^{-1}	-1.68×10^{-1}	0.821

Under redundancy, a coordinate can carry strictly positive marginal information about X while contributing nothing to the joint, so adding noise to it does not reduce $I(X;Z_{\mathrm{unimp}})$. The condition$I(X;Z_i \mid Z_{-i}) > 0$ in Claim 1 governs the rate.

F.3 Lemma 1 Verification

To verify Lemma 1, we evaluate a probe lower bound on the left-hand side, $\mathrm{CE}(Y \mid Z_{\mathrm{imp}}) - \mathrm{CE}(Y \mid Z_{\mathrm{imp}}, Z_{\mathrm{unimp}})$, against the closed-form Gaussian-channel upper bound of Theorem 1 on the right-hand side (Table 8).

The variational statistic $\mathrm{CE}(Y \mid Z_{\mathrm{imp}}) - \mathrm{CE}(Y \mid Z_{\mathrm{imp}}, Z_{\mathrm{unimp}})$ is confounded at these sample sizes because the probe sees 100–411 graphs against 128–256 features, so appending any block of columns degrades held-out cross-entropy whether or not those columns carry information. A Gaussian control of matched shape degrades it by a comparable amount in every setting we tested, confirming the effect is capacity rather than absence of signal. Reporting the naive statistic alone would therefore demonstrate overfitting, not conditional independence.

We instead shuffle Z_{unimp} across graphs. This destroys its association with Y while preserving its marginal distribution and the probe's parameter count exactly, so under the shuffled distribution the conditional mutual information is zero by construction. The statistic is

$$\delta \;=\; \mathrm{CE}\big(Y \mid Z_{\mathrm{imp}}, Z_{\mathrm{unimp}}^{\mathrm{shuf}}\big) \;-\; \mathrm{CE}\big(Y \mid Z_{\mathrm{imp}}, Z_{\mathrm{unimp}}\big),$$

the excess predictive gain of the true pairing over chance at matched capacity, with a permutation p-value over 25 draws. One regularization strength is selected by cross-validation on the full feature set and reused across the real, shuffled and Gaussian probes so that all three are matched. We report

Table 8: Conditional entropy of Y given the important subgraph representation Z_{imp}, with and without the unimportant subgraph representation Z_{unimp}. Δ denotes the increase in conditional entropy when Z_{unimp} is included.

Model / Dataset	$\text{CE}(Y \mid Z_{\text{imp}})$	$\text{CE}(Y \mid Z_{\text{imp}}, Z_{\text{unimp}})$	Δ
GSAT BA-2Motifs	0.667	2.141	+1.474
GSAT AIDS	0.323	0.486	+0.163
GSAT PROTEINS	0.642	1.692	+1.051
GSAT NCI1	0.651	0.782	+0.131
PGIB MUTAG	0.221	0.693	+0.472
PGIB BA-2Motifs	0.359	0.691	+0.332
PGIB AIDS	0.413	0.867	+0.454
PGIB PROTEINS	0.834	1.489	+0.655
PGIB NCI1	0.604	1.571	+0.967

δ as a permutation test rather than as a bound. Both probes are suboptimal, so δ estimates detectable leakage at this probe's power rather than the conditional mutual information itself.

Table 9: Lemma 1 on trained models. δ is the excess predictive gain of the true pairing over a capacity-matched permutation null, in nats; p is the permutation p-value over 25 draws. The right-hand column is the closed-form bound on $I(Z_{\text{unimp}}; X \mid Z_{\text{imp}})$ from Theorem 1. The inequality of Lemma 1 holds in every setting.

Model	Dataset	Graphs	δ	p	Lower bound	Upper bound
GSAT	BA-2Motifs	100	-0.122	0.920	0.000	0.0002
GSAT	AIDS	165	$+0.022$	0.320	0.000	0.136
GSAT	NCI1	410	-0.018	0.800	0.0012	0.006
GSAT	PROTEINS	110	-0.007	0.720	0.000	0.029
GSAT	MUTAG	1284	$+0.005$	0.040	0.002	0.003
PGIB	BA-2Motifs	100	$+0.208$	0.120	0.000	1.116
PGIB	MUTAG	296	$+0.038$	0.000	0.0238	4.259
PGIB	AIDS	200	$+0.021$	0.000	0.0185	3.473
PGIB	NCI1	411	$+0.005$	0.480	0.000	5.223
PGIB	PROTEINS	411	$+0.027$	0.040	0.0001	4.488

In six of the eight completed settings the true pairing is not distinguishable from the permutation null ($p > 0.05$). That is, no label information is detectable in the unimportant embeddings beyond what the important ones already supply. In two settings (PGIB on MUTAG and on AIDS) the true pairing does beat the null at $p = 0.000$, and in both the probe's cross-entropy genuinely improves when Z_{unimp} is appended ($0.360 \to 0.336$ and $0.070 \to 0.052$ nats). These two positives matter for interpreting the six negatives because it confirms that the test detects residual leakage where it exists, so the null results reflect its absence rather than a lack of statistical power.

Where leakage is detected it is small relative to the quantity that bounds it, $\delta = 0.038$ against an upper bound of 4.259 nats on MUTAG, and 0.021 against 3.473 on AIDS, i.e. under 1% in both cases. We therefore report the conditional independence used in Section 4.2.1 as approximate and bounded rather than exact. This is weaker than asserting $I(Z_{\text{unimp}}; Y \mid Z_{\text{imp}}) = 0$, and it is what the evidence supports. Lemma 1 is what makes the weaker statement sufficient, since it ties the residual to a term the module provably drives down.

We note that the GSAT–MUTAG test split contains a single class by construction, because explanation AUC is evaluated only on graphs carrying ground-truth edge labels, so the probe for that setting is fit on the validation split instead.

F.4 Proposition 2 Verification

We evaluate both bounds in a common 8-dimensional PCA subspace. For an orthonormal projection P, $PZ_i = Pf_i + \sqrt{t_i}P\epsilon_i$ with $P\epsilon_i \sim \mathcal{N}(0, I_k)$, so both bounds apply to $I(X; PZ_i)$, which is itself a lower bound on $I(X; Z_i)$ by the data processing inequality.

We evaluate the upper bound of Proposition 2 on node embeddings binned by importance quantile, at the σ used in training.

Table 10: Upper bound of Proposition 2 by importance quantile, in nats. Importantly, the bound increases with α within each row.

Model	Dataset	bin 1	bin 2	bin 3	bin 4
GSAT	BA-2Motifs	0.0002	0.0014	0.1496	0.1075
GSAT	MUTAG	0.0020	0.0031	0.0046	1.4986
GSAT	AIDS	0.0640	0.1912	1.0304	1.9400
GSAT	NCI1	0.0073	0.0126	0.1004	0.3593
GSAT	PROTEINS	0.0110	0.0388	0.0524	0.0643
PGIB	BA-2Motifs	2.3191	3.2810	3.5743	3.7971
PGIB	MUTAG	6.1663	8.8084	9.8369	10.5164
PGIB	AIDS	4.3396	5.9636	6.7558	6.8982
PGIB	NCI1	8.0119	10.1282	11.1238	11.7346
PGIB	PROTEINS	6.3391	10.2273	10.6086	11.2872

The bound increases monotonically with α in all PGIB and GSAT settings, consistent with the importance grading described in Proposition 2.

F.5 Effective rank of the bottleneck

Theorem 1, Claim 2, admits equality for injective ψ, so the MLP component is not the standalone compression mechanism. What it does control, in composition with the noise (component 1), is the effective rank of $\mathrm{Cov}(\tilde{Z})$ and hence the capacity bound. Table 11 measures this on randomly initialized $h \to w \to h$ blocks with $h = 64$, holding the output dimension fixed at 64 throughout so that any change is attributable to w alone.

Table 11: Effective rank and capacity as a function of bottleneck width, on randomly initialized blocks with output dimension fixed at 64. Participation ratio is $(\sum_j \lambda_j)^2 / \sum_j \lambda_j^2$; capacity is $\frac{1}{2}\sum_j \log(1 + \lambda_j/t)$ at $t = 0.5$.

Config.	h	Participation Ratio	Capacity
$h \to 64 \to h$	64	24.86	13.273
$h \to 32 \to h$	64	16.66	11.892
$h \to 16 \to h$	64	11.42	10.209
$h \to 8 \to h$	64	7.01	7.042
$h \to 4 \to h$	64	3.77	4.850

G Experimental Setup

Table 12: 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

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

Table 13: 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

Tables 14 and Table 15 give an overview of the dataset statistics used within the paper.

Table 14: 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 15: 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

H Explanatory Subgraph Visualization

I 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.

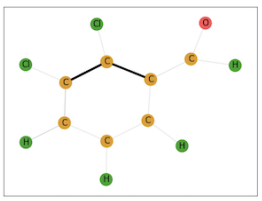
I.1 Node Drift

Table 16: 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.

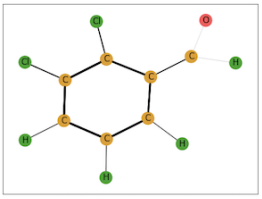
Architecture / Configuration	**MUTAG**	**BA-2Motifs**	**NCI1**	**PROTEINS**
GSAT	-0.02 ± 0.007	0.006 ± 0.006	-0.004 ± 0.006	0.002 ± 0.001
GSAT w/o IB Loss	-0.018 ± 0.007	-0.003 ± 0.006	0.028 ± 0.011	$\mathbf{0.004 \pm 0.013}$
GSAT w/o IB Loss + BottleneckMLP	$\mathbf{0.07 \pm 0.009}$	$\mathbf{0.02 \pm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

Table 17: 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.

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	$\mathbf{0.546 \pm 0.111}$	$\mathbf{0.214 \pm 0.049}$	$\mathbf{0.415 \pm 0.272}$	$\mathbf{0.937 \pm 0.563}$

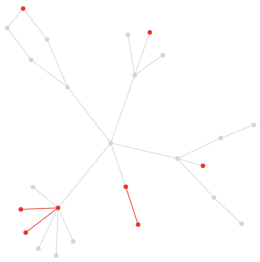


(a) GSAT

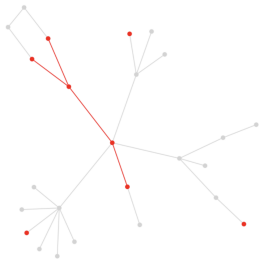


(b) GSAT + BottleneckMLP

Figure 4: 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 5: Visualization of explanatory subgraphs for BA-2Motifs dataset. (a) PGIB baseline, and (b) PGIB enhanced with BottleneckMLP. BottleneckMLP correctly identifies the cycle motif.

I.2 Convex Hull

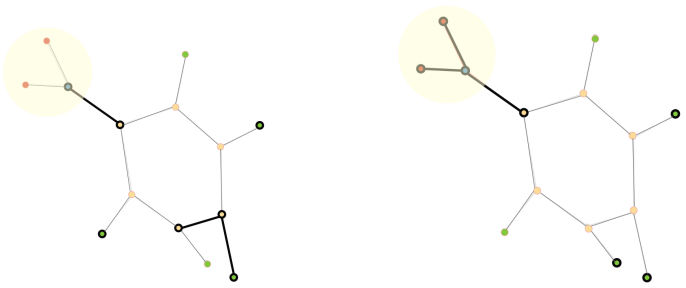
To further support our hypothesis that the BottleneckMLP influences representation structure locally—particularly around important nodes—we analyze the convex hull volumes and average linkage distances of node embeddings over training.

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 9, 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.

I.3 Node linkage distance

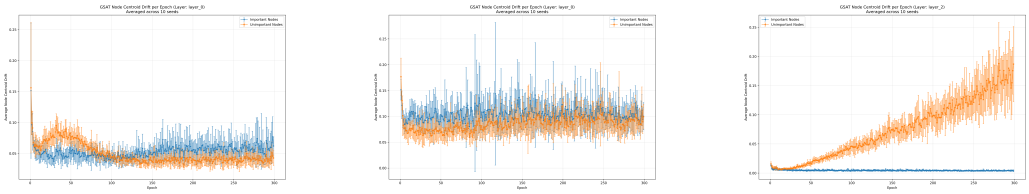
To complement this analysis, we compute the average linkage distance between node embeddings within each category. This metric quantifies the average pairwise distance between points in a cluster, offering an alternative view of intra- and inter-category embedding dynamics. The results, summarized in Table 19, 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 18. Notably, the largest inter-cluster distances are observed between Category 1 and Category 3, and between Category 2 and



(a) PGIB (b) PGIB + BottleneckMLP

Figure 6: 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.



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

Figure 7: 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 I).

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 18: Inter-Cluster Average Linkage for Categories and Layers. Categories 2 and 3 have higher inter-cluster average linkage distances than Category 1 node embeddings. This shows that the unimportant node embeddings become more dispersed from the important node embeddings.

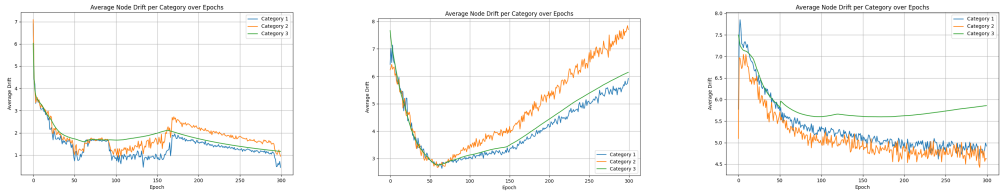
Layer	Cat1 - Cat2	Cat1 - Cat3	Cat2 - Cat3
Layer 1	14.955	22.934	23.155
Layer 2	285.214	450.272	451.345
Layer 3	236.287	383.437	383.836
Layer 4	409.867	568.589	568.324

Table 19: Intra-Cluster Average Linkage for Categories and Layers. Important node embeddings become more clustered, while unimportant node embeddings become more disperse.

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 19 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 10.

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



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

Figure 8: 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.

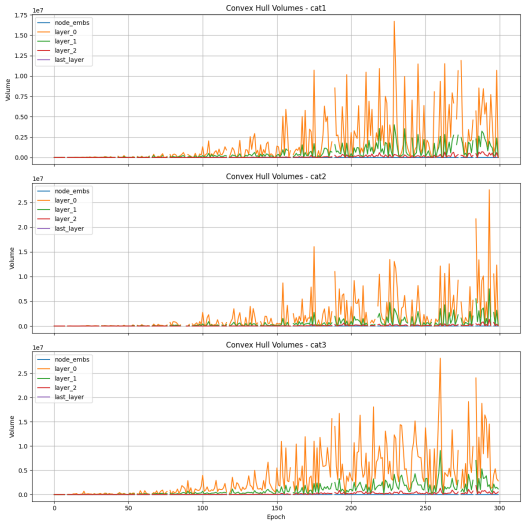


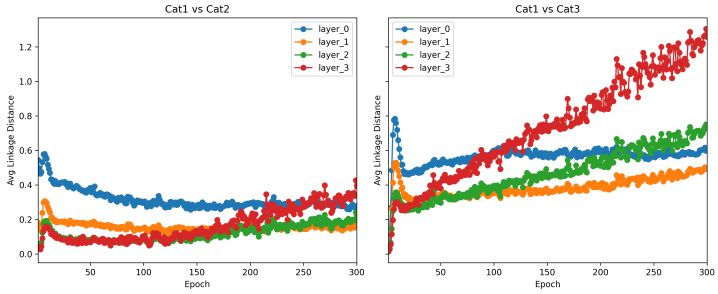
Figure 9: Convex hull volumes increase across categories. The x-axis spans epochs 0 to 300. The y-axis range is (0.0, 1.75e7) for Category 1 (top plot), and is (0.0, 3.0e7) for Categories 2 and 3 (middle and bottom plot). For all layers (most significantly for earlier layers), the convex hull volume increases across categories with Category 1 having the smallest convex hull volume, and Category 3 with the largest convex hull volume.

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.

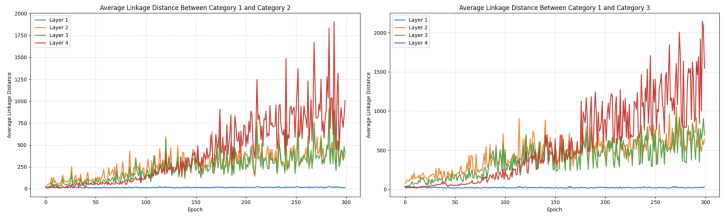
I.4 HSIC Results

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 20)



(a) GSAT node linkage visualization



(b) PGIB node linkage visualization

Figure 10: 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

Table 20: HSIC values across datasets for GSAT. Lower values indicate more independence. BottleneckMLP reports lowest HSIC values for 3/4 datasets.

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}	$\mathbf{3.89 \times 10^{-3}}$	$\mathbf{2.53 \times 10^{-3}}$	$\mathbf{6.37 \times 10^{-3}}$

J LINK PREDICTION

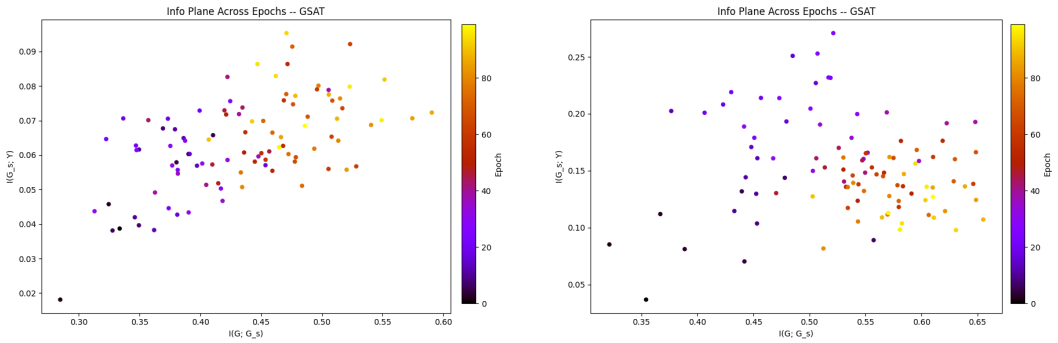
Table 4 Performance gain is largest on USLegis, while Wikipedia and CanParl have comparable or slightly better performance. Removing IB Loss boosts performance on Wikipedia and CanParl but reduces it on USLegis; adding BottleneckMLP recovers this loss and exceeds the baseline.

K AUC-ROC

Table 21: GSAT AUC-ROC. BottleneckMLP maintains explanation quality.

	MUTAG	BA-2Motifs
GSAT	0.995 $\pm$0.03	0.988$\pm$0.01
GSAT w/o IB Loss	0.998$\pm$0.04	0.996$\pm$0.00
GSAT w/o IB Loss + Bottleneck MLP	0.993$\pm$0.00	0.993$\pm$0.00

Table 21 shows that explanation AUC-ROC is maintained with BottleneckMLP. Figure 11 further reinforces this conclusion. In the information plane, GSAT with BottleneckMLP (Figure 11b) 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 11: 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.

L 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 22: PGIB/ GSAT Test Accuracy for various BottleneckMLP Architectures.

	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

M Generalization of BottleneckMLP across Graph Tasks and Explainer Types

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), post-hoc node classification Luo et al. (2020b), a non-IB-based ante-hoc explainer Wu et al. (2022), and non-IB-based post-hoc explainers Wang et al. (2023); Lu et al. (2024); Wang et al. (2021).

We note that the ablations removing size and entropy losses from PGExplainer, GNNExplainer, and analogous regularization terms from other non-IB-based methods are our own interventions, introduced to isolate the role of explicit loss-based compression and establish a fair basis for comparison with BottleneckMLP; they are not ablations reported in the original papers.

M.1 BottleneckMLP for Subgraph Recognition

For VGIB Yu et al. (2022), we removed the MI loss penalty term, and added BottleneckMLP. For VGIB in Table 23, the default model hidden dimension $h = 16$.

Table 23: VGIB Test Prediction Accuracy ± Std Dev. Variants of VGIB with BottleneckMLP have the highest accuracy across all datasets as compared with baseline VGIB, and VGIB without IB Loss.

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

M.2 BottleneckMLP for Post-hoc Node Classification

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 24 shows that we achieve better generalization and accuracy in node classification when BottleneckMLP is added.

Table 24: BottleneckMLP improves GNN Node Classification Accuracy

	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 24 shows that without the entropy loss, PGExplainer (Table 25) 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. For GNNExplainer (Table 26), BottleneckMLP effectively replaces the size and entropy loss.

Table 25: BottleneckMLP improves PGExplainer Explanation AUC-ROC for BA-Shapes and Tree-Grids. For TreeCycles, BottleneckMLP recovers the function of the Entropy Loss and achieved almost equivalent performance to the baseline.

	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

M.3 BottleneckMLP for Post-hoc Graph Classification

M.3.1 V-InFoR

V-InfoR Wang et al. (2023) approaches GNN explanation by first denoising corrupted graphs using a variational latent distribution of graph representations, then generating explanations by graph information bottleneck (GIB). We remove this GIB term and replace it with our BottleneckMLP. Table 27 demonstrates the superior performance of V-InFoR when we add BottleneckMLP on Fidelity, Sufficiency, Necessity, F_{ns} Score, and Mean Probability. BottleneckMLP increases performance on all metrics, with the exception for Prob-N for BA3, where the value is only 0.003 lower, and for Mean Probability on MUTAG.

M.3.2 ReFine

ReFine Wang et al. (2021) introduces a two-stage framework for graph explanation that combines global class-level reasoning with graph-specific refinement. In the pre-training stage, ReFine learns to distinguish graphs of different classes by optimizing a contrastive objective, enabling it to capture

Table 26: Performance of GNNExplainer variants across datasets (mean ± std).

Method	BAShapes	BACommunity	TreeCycles	TreeGrids
Original	0.504 ± 0.008	0.501 ± 0.009	0.499 ± 0.008	0.498 ± 0.013
No size or entropy loss	0.496 ± 0.012	0.503 ± 0.010	0.499 ± 0.020	0.499 ± 0.008
+ BottleneckMLP (fc-60-16)	0.500 ± 0.012	0.497 ± 0.009	0.493 ± 0.016	0.500 ± 0.006
+ BottleneckMLP (fc-60-16-8)	0.499 ± 0.011	0.503 ± 0.006	0.510 ± 0.014	0.499 ± 0.009

Table 27: Comparison of Fidelity and Explanation Metrics for BA3 and MUTAG on V-InFoR. Adding BottleneckMLP to V-InFoR increases performance on all five metrics on both datasets, with the exception of a slight decrease in Prob-N for BA3, and a decrease in Mean Probability with MUTAG.

Dataset	Metric	V-InFoR	V-InFoR w/o Info Loss	V-InFoR w/o Info Loss + BottleneckMLP
BA3	Fidelity	$\underline{0.5240} \pm 0.0096$	0.5192 ± 0.0129	$\mathbf{0.5308} \pm 0.0154$
	Prob-S	0.5105 ± 0.0123	$\underline{0.5218} \pm 0.0156$	$\mathbf{0.5236} \pm 0.0199$
	Prob-N	$\mathbf{0.6620} \pm 0.0010$	0.6595 ± 0.0035	$\underline{0.6617} \pm 0.0026$
	F_{ns} Score	0.5764 ± 0.0073	$\underline{0.5825} \pm 0.0103$	$\mathbf{0.5844} \pm 0.0120$
	Mean Probability	$\underline{0.6982} \pm 0.1981$	0.5104 ± 0.1091	$\mathbf{0.7607} \pm 0.1860$
MUTAG	Fidelity	$\underline{0.5718} \pm 0.0098$	0.5240 ± 0.0096	$\mathbf{0.5992} \pm 0.0159$
	Prob-S	$\underline{0.5718} \pm 0.0098$	0.5105 ± 0.0123	$\mathbf{0.5992} \pm 0.0159$
	Prob-N	$\underline{0.4749} \pm 0.0122$	0.4488 ± 0.0182	$\mathbf{0.6620} \pm 0.0010$
	F_{ns} Score	$\underline{0.5187} \pm 0.0088$	0.5130 ± 0.0148	$\mathbf{0.5764} \pm 0.0073$
	Mean Probability	$\mathbf{0.9675} \pm 0.0053$	0.4809 ± 0.1925	$\underline{0.7124} \pm 0.1736$

high-level, class-discriminative structural patterns. In the fine-tuning stage, these global cues are adapted to each individual graph, producing explanations that account for its local structure.

Table 28 shows that ReFine with BottleneckMLP has comparable performance to the baseline ReFine model on BA3 and MNIST datasets. BottleneckMLP improves ACC-AUC on MUTAG by 19.9%.

Table 28: Performance comparison (ACC-AUC and Mean Precision) across BA3, MNIST, and MUTAG for ReFine variants. BottleneckMLP improves performance most significantly on MUTAG.

Model Variant	BA3	MNIST	MUTAG
ReFine	**0.559**	$\underline{0.261}$	$\underline{0.694}$
ReFine w/o fidelity loss	0.505	0.199	0.664
ReFine w/ BottleneckMLP	$\underline{0.552}$	**0.263**	**0.832**

M.4 PROXYEXPLAINER

ProxyExplainer Chen et al. (2024) is a recent post-hoc method that leverages 'proxy graphs' generated by a Graph VAE to generate in-distribution explanations. Table 29 demonstrates the performance increase in Explanation AUC-ROC with the addition of BottleneckMLP.

Table 29: ProxyGraph Explanatation AUC and Fidelity+/-

Model	**Dataset**	**AUC**	**Fidelity+**	**Fidelity-**
ProxyExplainer	Fluoride Carbonyl	0.518 ± 0.081	0.050 ± 0.021	$\mathbf{0.409} \pm 0.066$
	Alkane Carbonyl	0.211 ± 0.075	0.112 ± 0.039	0.503 ± 0.184
	BA-2Motifs	$\mathbf{0.940} \pm 0.011$	$\mathbf{0.509} \pm 0.015$	0.201 ± 0.006
	MUTAG	$\mathbf{0.847} \pm 0.139$	0.714 ± 0.049	0.345 ± 0.127
ProxyExplainer + BottleneckMLP	Fluoride Carbonyl	$\mathbf{0.681} \pm 0.026$	$\mathbf{0.102} \pm 0.029$	0.633 ± 0.032
	Alkane Carbonyl	$\mathbf{0.527} \pm 0.332$	$\mathbf{0.775} \pm 0.142$	$\mathbf{0.253} \pm 0.176$
	BA-2Motifs	0.779 ± 0.032	0.500 ± 0.011	$\mathbf{0.000} \pm 0.000$
	MUTAG	0.822 ± 0.168	$\mathbf{0.721} \pm 0.036$	$\mathbf{0.357} \pm 0.112$

M.5 BottleneckMLP for Non-IB-based Ante-hoc Graph Classification

M.5.1 GOAt

GOAt Lu et al. (2024) learns stochastic, edge-level gates that selectively preserve only the most task-relevant subgraph during message passing, producing graph explanations directly from the forward pass. Table 30 demonstrates the superior performance of GOAt when we add BottleneckMLP on explanation quality, measured by Fidelity-, Fidelity+, and Sparsity. BottleneckMLP increases performance on all metrics, with the exception for the GOAt-GCN variant on BA-2Motifs. All other datasets and configurations have increased performance with BottleneckMLP.

Table 30: Explanation Quality Metrics (Fidelity-, Fidelity+, Sparsity) across GOAt Variants and Datasets. BottleneckMLP increases performance.

Model	Dataset	Fidelity-	Fidelity+	Sparsity
GOAt-GCN	BA-2Motifs	**0.001** $\pm$ 0.048	**0.542** $\pm$ 0.012	0.786 $\pm$ 0.076
	Mutagenicity	0.140 $\pm$ 0.040	0.653 $\pm$ 0.047	0.764 $\pm$ 0.013
	NCI1	0.086 $\pm$ 0.003	0.489 $\pm$ 0.072	0.835 $\pm$ 0.000
GOAt-GCN + BottleneckMLP	BA-2Motifs	0.056 $\pm$ 0.047	0.509 $\pm$ 0.005	**0.788** $\pm$ 0.041
	Mutagenicity	**0.073** $\pm$ 0.196	**0.702** $\pm$ 0.076	**0.786** $\pm$ 0.021
	NCI1	**0.004** $\pm$ 0.142	**0.563** $\pm$ 0.058	**0.848** $\pm$ 0.018
GOAt-GIN	BA-2Motifs	**0.001** $\pm$ 0.000	0.573 $\pm$ 0.004	0.836 $\pm$ 0.001
	Mutagenicity	0.436 $\pm$ 0.003	0.650 $\pm$ 0.003	0.808 $\pm$ 0.000
	NCI1	0.590 $\pm$ 0.110	0.744 $\pm$ 0.021	0.799 $\pm$ 0.001
GOAt-GIN + BottleneckMLP	BA-2Motifs	0.070 $\pm$ 0.000	**0.977** $\pm$ 0.001	**0.845** $\pm$ 0.002
	Mutagenicity	**0.020** $\pm$ 0.002	**0.692** $\pm$ 0.015	**0.843** $\pm$ 0.000
	NCI1	**0.500** $\pm$ 0.014	**0.783** $\pm$ 0.001	**0.810** $\pm$ 0.002
GOAt-SAGE	BA-2Motifs	0.479 $\pm$ 0.004	0.500 $\pm$ 0.004	0.813 $\pm$ 0.014
	Mutagenicity	0.040 $\pm$ 0.018	0.525 $\pm$ 0.005	0.812 $\pm$ 0.009
	NCI1	0.506 $\pm$ 0.174	0.674 $\pm$ 0.051	0.803 $\pm$ 0.013
GOAt-SAGE + BottleneckMLP	BA-2Motifs	**0.472** $\pm$ 0.004	**0.500** $\pm$ 0.001	**0.833** $\pm$ 0.001
	Mutagenicity	**0.013** $\pm$ 0.021	**0.572** $\pm$ 0.003	**0.823** $\pm$ 0.001
	NCI1	**0.391** $\pm$ 0.001	**0.728** $\pm$ 0.001	**0.808** $\pm$ 0.012

M.5.2 DIR

DIR (Wu et al., 2022) aims to learn causal rationales and confounding rationales to improve explanation. The method introduces an invariance loss which we consider as the "Info Loss" term to ablate in their method, as it is the analogous component of IB loss in DIR. Table 31 and Table 32 demonstrates the performance of DIR when we add BottleneckMLP on Causal ACC, Confounding ACC, Train ACC, Validation ACC, Test Precision@5, and Test MRR. BottleneckMLP increases Train ACC, Validation ACC, and Confounding ACC. In other most cases, DIR with BottleneckMLP slightly under-performs DIR, however, BottleneckMLP is able to recover the performance increase related to DIR when we remove the Info Loss terms, showing that it implicitly causes the same effects as does the explicit loss in DIR.

M.6 BottleneckMLP for Temporal Graph Classification

N 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 $P(G_S) = \sum_G P_\phi(G_S|G)P_G(G)$.

Table 31: DIR results comparing Original vs. BottleneckMLP across datasets (core accuracy metrics).

Dataset	Method	Causal ACC	Conf ACC	Train ACC	Val ACC
spmotif-0.9	DIR	**0.3921**±0.0607	0.3548±0.0216	0.7303±0.0974	0.4920±0.0963
	DIR w/o Info Loss	0.3488±0.0181	0.3386±0.0058	0.8780±0.0262	0.4167±0.1089
	BottleneckMLP	0.3505±0.0247	**0.3376**±0.0098	**0.9069**±0.0411	**0.4928**±0.1687
spmotif-0.7	DIR	**0.4089**±0.0570	0.3669±0.0079	0.6916±0.1210	0.5592±0.1478
	DIR w/o Info Loss	0.3708±0.0400	0.3582±0.0370	0.7291±0.0450	0.4888±0.1376
	DIR w/ BottleneckMLP	0.3794±0.0389	**0.3377**±0.0030	**0.8137**±0.1123	**0.6048**±0.2355
spmotif-0.5	DIR	**0.3918**±0.0444	0.3388±0.0100	0.6696±0.0661	0.6064±0.0935
	DIR w/o Info Loss	0.3590±0.0406	0.3419±0.0101	0.6280±0.2131	0.5347±0.2548
	DIR w/ BottleneckMLP	0.3907±0.0234	**0.3363**±0.0037	**0.7256**±0.0899	**0.6417**±0.1320
MNIST-75sp	DIR	0.2112±0.0297	0.1540±0.0209	0.2643±0.1018	0.2665±0.1018
	DIR w/o Info Loss	0.1572±0.0348	0.1655±0.0092	0.2600±0.0949	0.2600±0.0949
	DIR w/ BottleneckMLP	**0.2130**±0.0271	**0.1529**±0.0276	**0.3736**±0.1794	**0.3621**±0.1809
Graph-SST2	DIR	**0.8200**±0.0099	0.8169±0.0096	**0.9949**±0.0024	**0.9248**±0.0018
	DIR w/o Info Loss	0.8001±0.0087	0.8093±0.0095	0.9885±0.0096	0.9238±0.0095
	DIR w/ BottleneckMLP	0.8101±0.0174	**0.8000**±0.0178	0.9915±0.0055	0.9244±0.0053

Table 32: DIR results comparing Original vs. BottleneckMLP (test metrics).

Dataset	Method	Test Prec	Test MRR
spmotif-0.9	DIR	**0.2251**±0.0928	**0.3828**±0.1394
	DIR w/o Info Loss	0.1499±0.1234	0.2607±0.2175
	BottleneckMLP	0.1927±0.1432	0.3434±0.2640
spmotif-0.7	DIR	**0.2013**±0.1082	**0.3521**±0.1669
	DIR w/o Info Loss	0.1475±0.1204	0.2556±0.2103
	DIR w/ BottleneckMLP	0.1856±0.1292	0.3358±0.2392
spmotif-0.5	DIR	**0.2227**±0.1104	**0.3842**±0.1566
	DIR w/o Info Loss	0.1576±0.1341	0.2848±0.1987
	DIR w/ BottleneckMLP	0.1908±0.1270	0.3534±0.2316

Table 33: TGIB Classifier Accuracy.

	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	0.959	0.586	0.607

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 R.

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.

O 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:

$$F^- = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(y_i = \hat{y}_i) - \mathbb{I}(y_i = \hat{y}_i^k), F^+ = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(y_i = \hat{y}_i) - \mathbb{I}(y_i = \hat{y}_i^{1-k}), \tag{5}$$

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.

P 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.

The resultant figure is below, showing the LNSA for Category 1, 2, and 3 nodes.

Q Derivations

$$\begin{aligned} I(X;Z) &= \mathbb{E}_{X,Z}\left[\log \frac{p(x \mid z)}{p(x)}\right] \\ &= \mathbb{E}_{X,Z}\left[\log \prod_i \frac{p(x_i \mid \text{Patch}(x_i), z)}{p(x_i \mid \text{Patch}(x_i))}\right] = \mathbb{E}_{X,Z}\left[\sum_i \log \frac{p(x_i \mid \text{Patch}(x_i), z)}{p(x_i \mid \text{Patch}(x_i))}\right] \end{aligned} \tag{6}$$

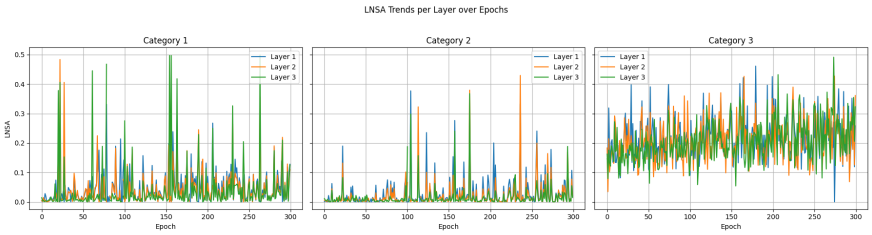


Figure 12: LNSA for PGIB + BottleneckMLP. Cat 1: important nodes. Cat 2: important nodes with unimportant neighbors. Cat 3: unimportant nodes. Cat 1 and Cat 2 embeddings are similar; Cat 3 has a higher mean LNSA as embedding structure evolves over epochs.

R PGIB

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

$$\mathcal{L}_{\text{CONT}} := -\frac{1}{n}\sum_{i=1}^{n}\log\frac{\exp(g(z_{G_i}, z_{G_j})/\tau)}{\sum_{k:z_k\notin P_{\text{sub}}}\exp(g(z_{G_i}, z_{G_k})/\tau)}. \tag{7}$$

S Running Time Analysis of BottleneckMLP

Adding BottleneckMLP has no effect on the running time of baseline models, as shown in Tables 34 and 35. BottleneckMLP's added cost is dominated by the MLP forward/backward pass, which scales as $O(N \cdot d \cdot h)$ per layer for N nodes, embedding dimension d, and hidden width h. The module's memory footprint is $O(N \cdot h)$ for activations, negligible relative to the GNN encoder's own message-passing memory. BottleneckMLP adds very limited computational or memory overhead to the underlying GNN/ explanation method and can thus handle high-dimensional graphs or large node-count networks accordingly.

Table 34: GSAT Training Mode Runtime (in minutes)

Mode	MUTAG	BA-2Motifs	NCI1	PROTEINS
GSAT	8.25 ± 0.70	4.03 ± 0.33	10.76 ± 0.96	7.23 ± 0.91
GSAT w/o IB Loss	9.15 ± 0.64	4.04 ± 0.18	10.26 ± 0.39	6.82 ± 0.95
GSAT w/o IB Loss + BottleneckMLP	9.47 ± 0.97	4.18 ± 0.15	10.66 ± 0.64	7.10 ± 1.04

Table 35: PGIB Training Runtime (in minutes)

Mode	MUTAG	BA-2Motifs	NCI1	PROTEINS
PGIB (Normal)	13.01 ± 1.36	5.46 ± 0.17	9.87 ± 1.38	91.78 ± 7.88
PGIB w/o IB Loss	15.90 ± 3.37	5.13 ± 0.28	9.90 ± 1.00	69.17 ± 11.97
PGIB w/o IB Loss + BottleneckMLP	9.95 ± 3.01	5.36 ± 0.74	12.75 ± 0.84	86.96 ± 14.07

T Convergence Analysis

PGIB with BottleneckMLP converges fastest, and most consistently as compared with original PGIB, and PGIB without IB Loss.

Hence, BottleneckMLP induces faster convergence via two mechanisms, corresponding to its two components.

(1) Layered MLP compression accelerates convergence relative to single-layer compression. The diffusion-time argument (Eq. 13, following the stochastic-relaxation analysis of (Tishby & Zaslavsky, 2015)) suggests this acceleration grows with the per-layer compression magnitudes; empirically we observe approximately 30–40% epoch reduction across datasets (Table 36)

With K hidden layers forming a Markov chain $X \to Z_1 \to \cdots \to Z_K$, each layer performs a partial compression, $\Delta I_X \approx \sum_k \Delta I_X^k$. The time required to achieve compression of magnitude ΔI_X via diffusion scales exponentially (Equation 8) as $\exp(\frac{\Delta I_X}{D})$, where D is a diffusion constant.

$$\exp\left(\sum_k \Delta I_X^k\right) \gg \sum_k \exp\left(\Delta I_X^k\right) \tag{8}$$

(2) BottleneckMLP effects the node embeddings by injecting noise scaled inversely by node importance, making important node embeddings more stable, and causing unimportant node embeddings to drift. Hence, this additionally accelerates the representation dynamics of the node embeddings over training epochs, leading to lower epochs needed for model convergence (Table 36), and improved performance.

Table 36: Comparison of convergence epoch for original PGIB, PGIB w/o IB Loss, and PGIB w/o IB Loss + BottleneckMLP configurations across datasets. All results are averaged over 10 seeds. The maximum number of epochs was 500.

Dataset/ Mode	PGIB	PGIB w/o IB Loss	PGIB w/o IB Loss + BottleneckMLP
BA 2Motifs	239 ± 136	256 ± 167	$\mathbf{149} \pm 54$
MUTAG	341 ± 134	412 ± 112	$\mathbf{240} \pm 65$
NCI1	216 ± 148	219 ± 162	$\mathbf{160} \pm 172$
PROTEINS	500 ± 0	500 ± 0	$\mathbf{358} \pm 174$
FluorideCarbonyl	396 ± 136	250 ± 177	$\mathbf{246} \pm 55$
AlkaneCarbonyl	397 ± 155	246 ± 135	$\mathbf{250} \pm 77$
Benzene	346 ± 189	260 ± 189	$\mathbf{192} \pm 156$

U Comparative Analysis of Information-Plane Dynamics Across Models

To understand how architectural choices influence information flow in graph neural networks, we compare the information-plane trajectories of three model variants. For each model, we plot $(I(X; Z_i))$ versus $(I(Z_i; Y))$ across training epochs (color-coded from purple to yellow). We show that the characteristic IB curve does not arise naturally in GNNs, but does arise with the addition of architectural constraints via BottleneckMLP.

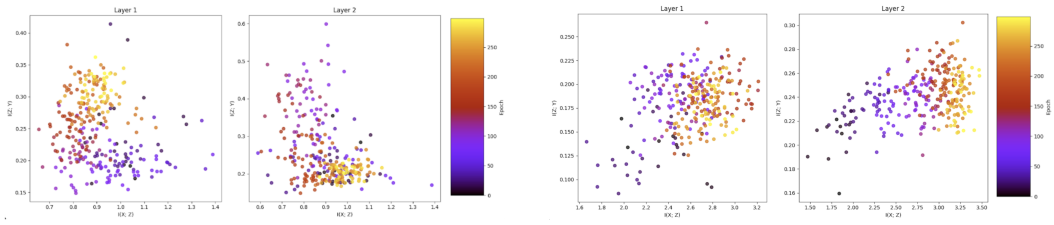
Baseline Model. Figure 13a and 14a shows that the baseline GNN does not exhibit the expected IB behavior. Across both layers, the scatter of points remains diffuse with no clear epoch-wise trajectory. The mutual information with the input, $(I(X; Z_i))$, remains relatively high throughout training, while $(I(Z_i; Y))$ does not show late-stage concentration. Early and late epochs are mixed together, indicating little systematic change in information content over training. These results confirm that, in standard GNN architectures, compression does not naturally emerge even with cross-entropy optimization.

Baseline Model w/o Info Loss. When removing the explicit information loss term (Figure 13b and 14b), the model retains even more information about the inputs, reflected by substantially higher $(I(X; Z_i))$ values compared to the baseline. While there is a weak upward trend in $(I(Z_i; Y))$ for later epochs, there is still no evidence of compression in $(I(X; Z_i))$.

BottleneckMLP. In contrast, the with BottleneckMLP, (Figure 3) displays a pronounced IB structure. Early layers behave similarly to those in the baseline model, but beginning around Layer 3, clear compression in $(I(X; Z_i))$ emerges as training progresses. Points transition from right (high

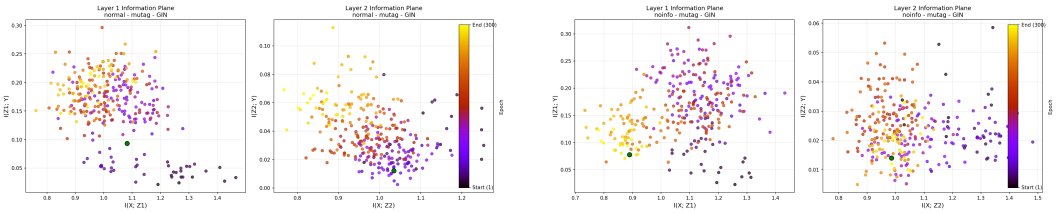
($I(X; Z_i)$)) to left (compressed representations) while ($I(Z_i; Y)$) becomes more concentrated. By Layers 4 and 5, the classical IB trajectory is clearly visible: the model discards redundant input information while retaining label-relevant structure. This trend aligns with theoretical expectations and with our discussion in Section 4.1.

Collectively, these observations demonstrate that GNNs do not naturally follow IB dynamics, even under standard cross-entropy optimization. The IB curve only emerges when compression is architecturally enforced through an explicit bottleneck.



(a) PGIB (normal) (b) PGIB (no-info)

Figure 13: PGIB information plane $I(X; Z_i)$ vs. $I(Z_i; Y)$ over baseline GNN layers with (left) and without(right) Info Loss, across epochs (purple to yellow). Information curve Tishby & Zaslavsky (2015) does not appear.



(a) GSAT (normal) (b) GSAT (no-info)

Figure 14: GSAT information plane $I(X; Z_i)$ vs. $I(Z_i; Y)$ over baseline GNN layers with (left) and without(right) Info Loss, across epochs (purple to yellow). Information curve Tishby & Zaslavsky (2015) does not appear.

V PSEUDOCODE

We provide the pseudocode for the importance-weighted noise injection of BottleneckMLP during the training of each explainer model in Algorithm 1.

Algorithm 1: Importance-Weighted Noise Injection Training Loop

Input: Graph G, Label y
Input: Classifier f_θ, subgraph extractor g_ϕ, noise scale σ
Initialize parameters ϕ, θ
for *epoch* $\leftarrow 1$ **to** E **do**
  $Z \leftarrow f(G)$; `// Initial node embeddings`
  $\alpha \leftarrow \mathrm{Sig}(g_\phi(Z))$; `// Importance score generation`
  Sample $\epsilon \sim \mathcal{N}(0, I_d)$
  $S \leftarrow \dfrac{\sigma}{\alpha}$
  $\tilde{Z} \leftarrow Z_B + \sqrt{S} \odot \epsilon$; `// Importance-weighted noise`
  $\hat{Y} \leftarrow h_\psi(\tilde{Z}, \alpha)$
  $\mathcal{L} \leftarrow \mathrm{CrossEntropy}(Y_B, \hat{Y})$;
  Update ϕ, θ to minimize $\mathcal{L}$

W GNN BACKBONES

We ran additional experiments with various GNN backbones to demonstrate the general applicability of BottleneckMLP. These results are shown for test accuracy and explanation AUROC for GSAT.

Table 37: Accuracy and AuROC results for different backbone models and BottleneckMLP variants.

Model Variant	MUTAG (Acc)	NCI1 (Acc)	PROTEINS (Acc)	MUTAG (AuROC)
GAT (normal)	0.893 ± 0.026	0.726 ± 0.016	$\mathbf{0.893 \pm 0.026}$	$\mathbf{0.690 \pm 0.304}$
GAT (noinfo)	0.892 ± 0.025	0.727 ± 0.015	$\underline{0.892 \pm 0.025}$	0.431 ± 0.301
GAT + BottleneckMLP (64–48–32)	$\mathbf{0.913 \pm 0.021}$	$\mathbf{0.748 \pm 0.026}$	0.725 ± 0.046	$\underline{0.482 \pm 0.390}$
GAT + BottleneckMLP (64–16)	$\underline{0.900 \pm 0.020}$	$\underline{0.743 \pm 0.025}$	0.733 ± 0.041	0.478 ± 0.317
GCN (normal)	0.885 ± 0.027	0.726 ± 0.014	$\underline{0.713 \pm 0.048}$	0.534 ± 0.369
GCN (noinfo)	0.893 ± 0.033	$\underline{0.727 \pm 0.015}$	0.702 ± 0.047	$\underline{0.571 \pm 0.346}$
GCN + BottleneckMLP (64–48–32)	$\underline{0.908 \pm 0.027}$	$\mathbf{0.743 \pm 0.013}$	$\mathbf{0.746 \pm 0.043}$	0.520 ± 0.403
GCN + BottleneckMLP (64–16)	$\mathbf{0.915 \pm 0.021}$	$\mathbf{0.743 \pm 0.013}$	$\mathbf{0.746 \pm 0.043}$	$\mathbf{0.836 \pm 0.129}$
PNA (normal)	$\underline{0.939 \pm 0.029}$	0.734 ± 0.026	0.640 ± 0.061	$\mathbf{0.993 \pm 0.005}$
PNA (noinfo)	$\mathbf{0.947 \pm 0.017}$	$\mathbf{0.781 \pm 0.028}$	$\underline{0.705 \pm 0.093}$	$\underline{0.972 \pm 0.008}$
PNA + BottleneckMLP (64–16)	$\underline{0.939 \pm 0.015}$	0.749 ± 0.026	$\mathbf{0.725 \pm 0.052}$	0.965 ± 0.010
PNA + BottleneckMLP (64–48–32)	0.935 ± 0.021	$\underline{0.760 \pm 0.025}$	0.695 ± 0.069	0.959 ± 0.012

X HYPERPARAMETER ANALYSIS

We include analysis of the hyperparameter σ which parametrizes the maximum gaussian noise variance in our method. We tune this for different datasets across different methods. The delay is currently set as 50 epochs (chosen so that importance scores have stabilized, assessed via validation-set consistency of the top-k node ranking across consecutive epochs). Figure 15 shows that GSAT classifier accuracy is not too sensitive to different σ values, and we find the sweet spot for accuracy with grid search. Explanation AUC-ROC drops for too high values of σ in the early stages of training, which detriments learning. For PROTEINS, we see a continuous increase in performance when σ is increased. MUTAG and NCI1 datasets are less sensitive to these changes, while BA2Motifs needs more tuning to find the sweet spot. This is potentially due to the synthetic nature of the dataset, and the randomness in the distribution of the edges that don't belong to the base motif.

Figures 17 and 18 show the sensitivity analysis for σ on PGIB with datasets MUTAG and BA-2Motifs. The optimal value of σ varies across datasets and methods. σ increases classifier and explanation performance, however, too high values of σ will introduce too much noise during early training, as reflected in the performance drop at higher values of σ.

Y SPARSITY

We use the default top-k protocol followed by each baseline method. That is, for each method, we retain the same k used by the original explainer (e.g., k = 5 or k = 10 depending on the dataset/benchmark). This ensures that BottleneckMLP is evaluated under the same sparsity budget as the original method, making the comparison fair and directly aligned with prior work.

While standard IB adaptively balances compression and prediction via the IB trade-off parameter (β), BottleneckMLP does not include an explicit sparsity-controlling term. Instead, sparsity emerges implicitly from the entropy separation induced by the noise injection: important nodes stabilize in low-entropy clusters, while unimportant nodes drift toward Gaussianized, high-entropy distributions.

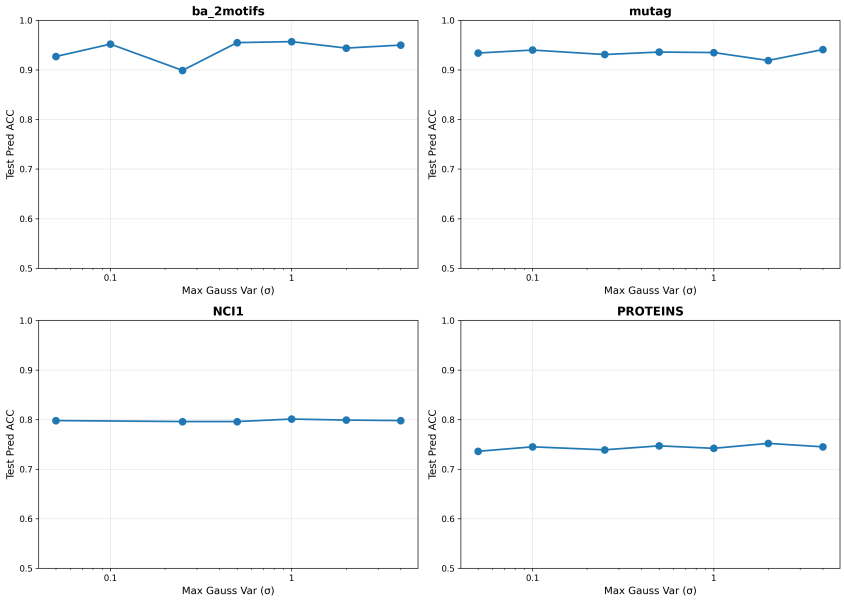


Figure 15: GSAT Classifier Accuracy for Different Sigma Values

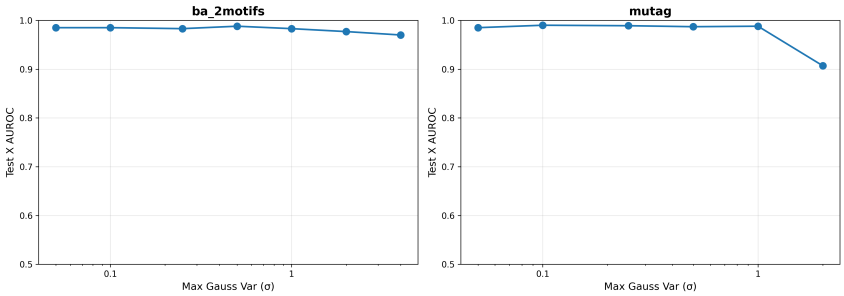


Figure 16: GSAT Explanation AUC-ROC for Different Sigma Values

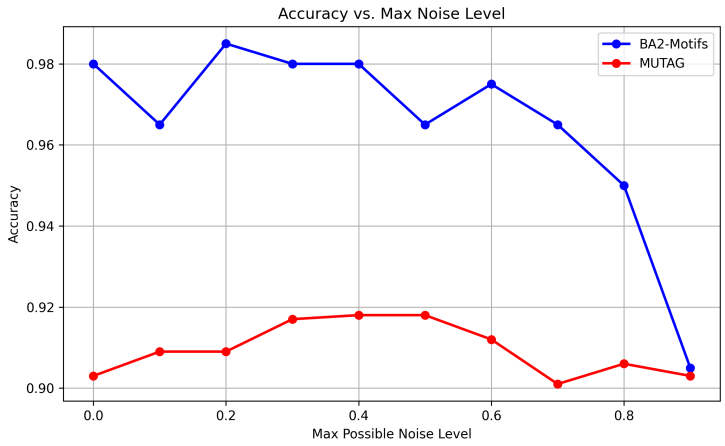
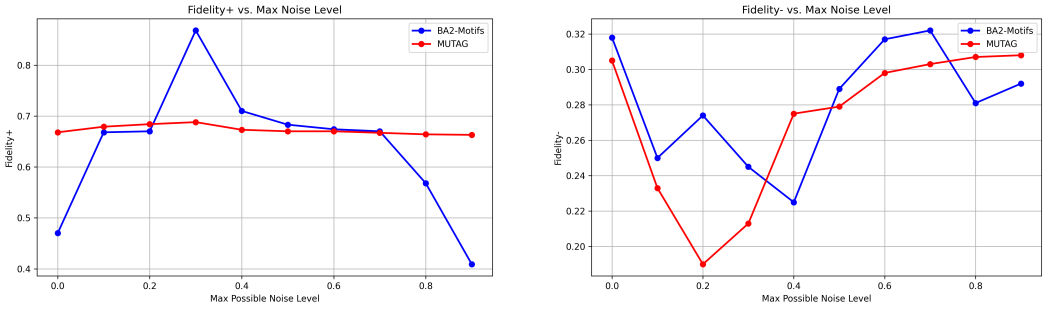


Figure 17: PGIB Classifier Accuracy for Different Sigma Values



(a) PGIB Fidelity+ for different σ values

(b) PGIB Fidelity- for different σ values

Figure 18: PGIB Fidelity+ and Fidelity- as a function of noise level σ.

In Figure 19, we show the effect of differing sparsity values, k, where k is the percent of edges in the explanatory subgraph.

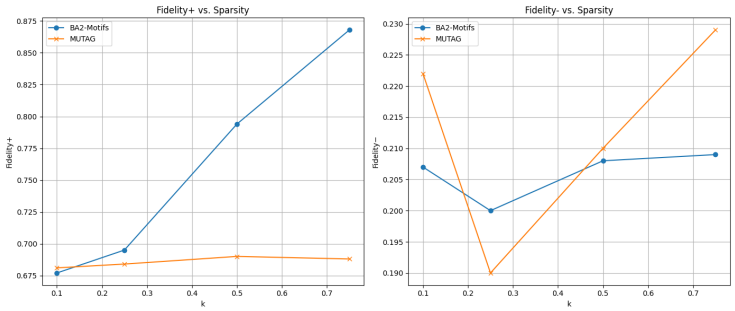


Figure 19: PGIB Fidelity+/ Fidelity- for Different Sparsity Values

Z Robustness to Importance Score Quality

BottleneckMLP scales noise inversely with node importance scores α_i, which raises a natural question: if these scores are unreliable, particularly early in training or under a weak base explainer, does noise injection incorrectly target task-relevant nodes? First, we note that we do not inject any noise until a defined minimum number of epochs, and we can scale the temperature over epochs. Second, we have shown that in any case, BottleneckMLP increases classification accuracy across all datasets and methods.

Assumption and its scope. We state explicitly the assumption underlying our mechanism: importance scores α_i are positively correlated with ground-truth node relevance, at least in expectation over training. In practice, scores are initially noisy and improve as the explainer trains; during early epochs, BottleneckMLP's MLP compression provides inherent regularization regardless of score quality, since architectural compression acts on all embeddings uniformly before subgraph extraction. The compression component of BottleneckMLP imposes representational structure even when importance scores are imperfect, providing a degree of robustness beyond score quality alone. As the base explainer improves over training, the noise-injection component becomes increasingly well-targeted, creating a self-reinforcing dynamic that accelerates convergence (Table 36).

Robustness to uninformative explainers. To directly test sensitivity to score quality, we simulate a maximally unreliable explainer by freezing the subgraph extractor's MLP at random initialization, so that importance scores α_i carry no learned signal. Table 38 reports classifier accuracy under this condition. Despite receiving entirely uninformative guidance, BottleneckMLP consistently outperforms both the standard baseline and the no-info ablation, most strikingly on BA-2Motifs (0.994

vs. 0.708 baseline). This demonstrates that the MLP compression component alone imposes sufficient structure to filter noise and preserve task-relevant information, even in the absence of reliable importance scores.

Table 38: Classifier accuracy under a randomly initialized (frozen) subgraph extractor, simulating a completely uninformative importance-scoring mechanism. BottleneckMLP outperforms all baselines in every setting.

Configuration	MUTAG	BA-2Motifs	PROTEINS	NCI1
Normal (Baseline)	0.887	0.708	0.682	0.652
w/o IB Loss	0.894	0.716	0.673	0.635
w/o IB Loss + Ours	**0.923**	**0.994**	**0.709**	**0.684**

Robustness to spurious correlations. We further evaluate resilience to unreliable signal under distribution shift using the SP-Motifs dataset Wu et al. (2022), a synthetic out-of-distribution benchmark in which graph motifs are spuriously correlated with background substructures at varying bias strengths $b \in \{0.5, 0.7, 0.9\}$. A higher b means the background is a stronger confound, making importance scores from a naive explainer increasingly misleading. As shown in Table 39, BottleneckMLP achieves the highest accuracy at every bias level, and its advantage over the baselines grows as spurious signal intensifies. This is the opposite of what would be expected if the method were critically dependent on score reliability. Architectural compression prevents spurious background features from dominating the classifier.

Table 39: Accuracy on SP-Motif across bias strengths b. Higher b indicates stronger spurious correlation between background structures and labels. BottleneckMLP's advantage increases as the spurious signal intensifies, demonstrating robustness beyond dependence on importance score quality.

Configuration	$b = 0.5$	$b = 0.7$	$b = 0.9$
GSAT	0.531	0.503	0.462
w/o IB Loss	0.398	0.516	0.435
w/o IB Loss + Ours	**0.545**	**0.519**	**0.498**

Consistency across backbone architectures. Table 37 further supports low sensitivity to the importance-generation mechanism: across three GNN backbones with substantially different inductive biases (GAT, GCN, PNA), BottleneckMLP consistently improves or maintains accuracy and explanation AUC-ROC, confirming that its benefits do not rely on any particular architectural choice for score generation.

Taken together, these results demonstrate that BottleneckMLP's primary contribution extends beyond modular improvement of existing explainers. By grounding graph representation learning in IB theory and providing an architecture-driven realization of the IB principle, BottleneckMLP achieves the desired compression and fitting phases without variational bounds. The consistent empirical improvements across post-hoc and ante-hoc methods, graph and node classification, and subgraph recognition are best understood as a consequence of this theoretical grounding rather than mere score-guided noise injection.

AA Comparison with Dropout and Noise-Injection Baselines

AA.1 Comparison with Dropout as a Noise-Injection Baseline

Information Dropout Achille & Soatto (2017) and VIB Alemi et al. (2018) also use noise injection as an implicit IB mechanism, but rely on input-conditioned variance (typically learned end-to-end) rather than scaling by an explainer's importance score. BottleneckMLP differs in two respects: (i) it grounds the noise scale in the host explainer's importance signal α, making the noise importance-aware rather than input-conditioned; (ii) it adds the architectural MLP bottleneck, which provides

the dominant compression effect (Appendix Y). The dropout comparison above shows that uniform noise without these refinements actively degrades performance.

Dropout differs from BottleneckMLP in two fundamental ways that explain its failure as a substitute. First, dropout applies noise uniformly across all nodes regardless of importance, whereas BottleneckMLP scales noise inversely to α_i, inducing the entropy separation between important and unimportant node representations that drives both explanation quality and task performance (Section 4.1). Second, the MLP compression layers in BottleneckMLP constitute an independent architectural bottleneck in addition to the noise-injection component. Table 40 confirms this empirically as dropout degrades performance on several datasets relative to the baseline, whereas BottleneckMLP consistently improves over all baselines across both explainers.

Table 40: Classifier accuracy comparing BottleneckMLP against dropout as a noise-injection baseline for GSAT and PGIB. Dropout applies uniform noise and degrades performance on several datasets; BottleneckMLP consistently improves over all baselines.

Method	MUTAG	BA-2Motifs	NCI1	PROTEINS
GSAT	$0.909 \pm .033$	$0.994 \pm .006$	$0.689 \pm .017$	$0.681 \pm .036$
GSAT w/o IB Loss	$0.935 \pm .043$	$0.770 \pm .224$	$0.743 \pm .024$	$0.706 \pm .040$
GSAT + Dropout	$0.933 \pm .048$	$0.546 \pm .126$	$0.666 \pm .025$	$0.682 \pm .049$
GSAT w/o IB Loss + BottleneckMLP	$\mathbf{0.949 \pm .010}$	$\mathbf{1.000}$	$\mathbf{0.802 \pm .019}$	$\mathbf{0.745 \pm .048}$
PGIB	$0.904 \pm .010$	$0.628 \pm .171$	$0.729 \pm .024$	$0.729 \pm .024$
PGIB w/o IB Loss	$0.916 \pm .011$	$0.896 \pm .145$	$0.774 \pm .014$	$0.763 \pm .022$
PGIB + Dropout	0.800	0.450	0.618	0.545
PGIB w/o IB Loss + BottleneckMLP	$\mathbf{0.925 \pm .009}$	$\mathbf{0.963 \pm .016}$	$0.753 \pm .019$	$\mathbf{0.792 \pm .018}$

The failure of dropout is most pronounced on BA-2Motifs, where GSAT + Dropout collapses to 0.546 against a baseline of 0.994, and PGIB + Dropout falls to 0.450 against 0.628. This is precisely the dataset where the IB compression objective matters most (Table 2), confirming that undifferentiated noise injection actively destroys the structured representations needed for accurate subgraph identification. BottleneckMLP achieves the opposite effect on the same dataset, recovering and surpassing baseline performance, because its noise is targeted and unimportant nodes are perturbed toward high-entropy distributions while important nodes retain low-entropy, structured embeddings.

AA.2 Other Noise-Injection Baselines

We also report results with no noise, uniform noise, shuffled importance scores, and residual skips below in Tables 42 and 41, respectively.

Table 41: Comprehensive performance comparison across multiple datasets.

Method	MUTAG	BA-2Motifs	NCI1	PROTEINS	Benzene	Alk.Carbonyl	Flu.Carbonyl
Ours	0.949 ± 0.010	1.000 ± 0.000	0.802 ± 0.019	0.745 ± 0.048	1.000 ± 0.000	0.995 ± 0.006	0.987 ± 0.013
No noise	0.933 ± 0.027	0.867 ± 0.152	0.762 ± 0.022	0.713 ± 0.046	1.000 ± 0.000	0.996 ± 0.006	0.987 ± 0.017
Uniform noise	0.936 ± 0.017	0.844 ± 0.191	0.769 ± 0.016	0.722 ± 0.038	1.000 ± 0.000	0.945 ± 0.026	0.964 ± 0.005
Shuffled importance	0.917 ± 0.024	0.824 ± 0.119	0.772 ± 0.020	0.705 ± 0.036	1.000 ± 0.000	0.946 ± 0.034	0.964 ± 0.002
Residual skip	0.927 ± 0.035	0.819 ± 0.177	0.765 ± 0.018	0.711 ± 0.032	1.000 ± 0.000	0.990 ± 0.011	0.968 ± 0.003

All parameter-matched controls underperform BottleneckMLP on all datasets (with the exception of Alk. Carbonyl under no noise), and that noise-based ablations degrade explanation performance (Table 41).

AB BottleneckMLP Component Ablation

We study the role of the scaled noise component in BottleneckMLP by comparing fidelity with and without noise. Table 42 shows that incorporating noise consistently improves explanation quality across datasets.

The noise-augmented variant achieves higher Fid^+ on all datasets, indicating better preservation of important substructures. At the same time, it yields lower (or comparable) Fid^- in most cases, suggesting more effective removal of uninformative components.

These results support the view that noise acts as an implicit regularizer, encouraging the bottleneck to discard spurious correlations while retaining task-relevant signals. Without noise, the model exhibits weaker separation between important and unimportant features, leading to degraded fidelity. As such, the scaled noise component plays a critical role in stabilizing and sharpening explanations.

We observe that BottleneckMLP without an explicit IB objective consistently achieves the highest or comparable Fid$^+$ across all datasets, indicating that the architecture alone induces sufficient compression for faithful explanations. Incorporating an explicit IB objective (PGIB) does not yield further improvements. These results suggest that architectural compression via BottleneckMLP subsumes the role of an explicit IB objective.

The MLP compression block accounts for the majority of the gain in predictive accuracy and provides baseline fidelity recovery. Adding the importance-scaled noise yields a further 5-15% relative improvement in Fid+ and Fid− across datasets, and reduces the epochs to convergence by 30-40% (Appendix T). Combining BottleneckMLP with the original IB Loss yields no additional benefit beyond IB Loss alone, supporting the substitution thesis: BottleneckMLP and explicit IB Loss occupy the same mechanistic slot.

Table 42: Fidelity using BottleneckMLP with and without scaled noise component, and full baseline model (PGIB) with BottleneckMLP.

Dataset	**PGIB w/o IB + BottleneckMLP**		**PGIB w/o Info Loss + BottleneckMLP (no noise)**		**PGIB + BottleneckMLP**	
	Fid$^+$	Fid$^-$	Fid$^+$	Fid$^-$	Fid$^+$	Fid$^-$
MUTAG	**0.76**±0.07	**0.38**±0.25	0.70±0.04	0.41±0.23	0.75±0.12	0.40±0.20
BA-2Motifs	**0.98**±0.08	**0.40**±0.24	0.94±0.12	0.47±0.16	0.98±0.09	0.43±0.22
NCI1	**0.77**±0.16	0.48±0.21	0.67±0.03	**0.48**±0.21	0.75±0.12	0.47±0.20
PROTEINS	**0.66**±0.02	0.59±0.09	0.64±0.02	0.59±0.07	0.65±0.02	**0.57**±0.08

AC EDGE VALIDATION AS MI ESTIMATOR

All MI results use EDGENoshad et al. (2018). Appendix I.4's HSIC is a fully estimator-free cross-check, agreeing qualitatively with EDGE on 3/4 datasets. To rule out estimator artifacts and validate our information-plane conclusions, we benchmarked EDGE against an analytical ground truth, and cross-verified our training trends using an independent neural estimator (MINE).

We tested the estimators on a synthetic bottleneck with known MI. We define $X = [X_{signal}, X_{noise}]$, $Z = X_{signal} + \beta \cdot X_{noise} + \sigma\epsilon$, and $Y = X_{signal}$. Lowering the value of β simulates increasing compression. Both estimators recover the trends.

β	True $I(X;Z)$	EDGE $I(X;Z)$	MINE $I(X;Z)$	True $I(Z;Y)$	EDGE $I(Z;Y)$	MINE $I(Z;Y)$
1.0	1.099	1.342	1.089	0.294	0.431	0.300
0.5	0.896	1.155	0.887	0.549	0.686	0.549
0.1	0.809	0.950	0.814	0.789	0.909	0.787
0.0	0.805	0.922	0.808	0.805	0.922	0.808

MINE corroborates the late compression trend but does not resolve the early fitting phase because of critic underfitting.

Table 43: Tracking MI in GSAT training using EDGE and MINE.

Epoch	EDGE $I(X;Z)$	MINE $I(X;Z)$	EDGE $I(Z;Y)$	MINE $I(Z;Y)$
1	1.025	0.686	0.031	-0.003
50	1.104	0.524	0.014	+0.008
150	1.116	0.492	0.035	+0.008
300	0.866	0.484	0.061	+0.012