Test-Time Graph Recalibration: Enhancing Robust Zero-Shot Inference for Graph Foundation Models

Chunchun Chen[1], **Zhen Luo**[2], **Xing Wei**[3], **Yuxing Zhang**[4], **Xiaofeng Cao**[5]
Rui Fan[3,1], **Ambuj Singh**[6], **Wei Ye**[3,1*]
[1]Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University, China
[2]Guohao School, Tongji University, China
[3]College of Electronic and Information Engineering, Tongji University, China
[4]School of Computer Science, Shanghai Jiao Tong University, China
[5]School of Computer Science and Technology, Tongji University, China
[6]Department of Computer Science, University of California, Santa Barbara, CA, USA
{c2chen, yew}@tongji.edu.cn

Abstract

This work investigates robust graph learning within graph foundation models. Prior studies primarily rely on data-centric structural purification or adversarial augmentation during training to achieve robustness. However, in practical zero-shot inference, such methods remain vulnerable to unseen adversarial structures because of static defenses and prohibitive retraining costs. We propose Test-time gRaph recAlibration for enhancing robust zero-shot inferenCE (TRACE). TRACE selectively learns a structural *anti-attack* directly on noisy graphs during inference, neutralizing adversarial perturbations while preserving clean-graph performance. Specifically, TRACE introduces a non-parametric diagnostic metric based on smoothed spectral entropy to quantify structural-semantic misalignment induced by adversarial attacks, providing an adaptive recalibration trigger. Furthermore, a uniformity-guided objective leverages semantic anchors from pretrained text encoders to guide the distorted graph back to the clean manifold. To ensure efficiency for sparse graph encoders, a first-order gradient-based edge-flipping strategy reconstructs the optimal graph structure directly in the discrete domain. Extensive experiments across benchmark datasets and graph attacks demonstrate TRACE's superiority over state-of-the-art baselines.

1 Introduction

Text-attributed graphs (TAGs) are prevalent across numerous real-world applications, from academic citation networks [1, 2] to e-commerce platforms [3]. Consequently, these structures have garnered interest from academia and industry [4, 5, 6]. Following the success of vision-language foundation models [7, 8, 9, 10, 11, 12, 13], recent research [14, 15] has focused on developing graph foundation models to enhance zero-shot generalization capabilities on non-Euclidean data. However, empirical studies [16, 17, 18] indicate that even minor structural perturbations can significantly alter the predictions of these graph foundation models, exposing a vulnerability inherent in most neural systems. In safety-critical domains such as finance [19] or healthcare [20], although it is difficult for adversaries to directly breach secure systems to tamper with user profiles, they can still manipulate model behavior by injecting spurious links or concealing legitimate interactions. Therefore, ensuring the robustness of graph foundation models has emerged as a paramount concern.

*Corresponding author.

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

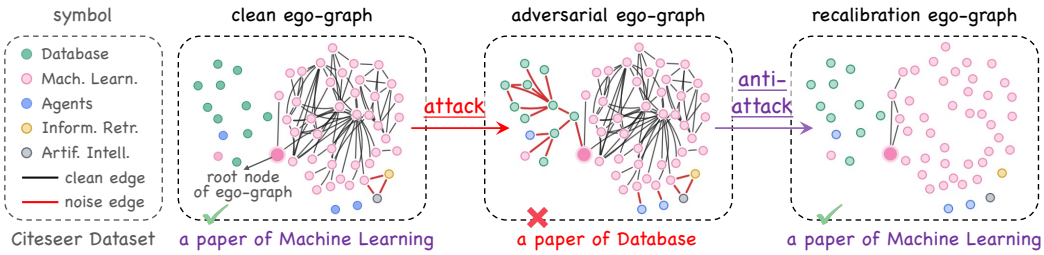

Figure 1: Empirical case study of adversarial attack and anti-attack recalibration on the CiteSeer dataset. The clean ego-graph, labeled *Machine Learning*, maintains intrinsic structural-semantic consistency. Topological attacks add noise edges, forcing misclassification as *Database*. Our framework proactively recalibrates via reverse optimization to prune adversarial edges and restore the correct semantic prediction.

In the literature, existing robust graph learning approaches are broadly categorized into data-centric structural purification [21, 22, 23, 24] and robust training-based methods [25, 26, 27, 28, 16, 17, 18]. Specifically, the former paradigm emphasizes the restoration of graph integrity by employing various pre-processing techniques to identify and remove adversarial edges before the model is formally deployed. Conversely, the latter focuses on enhancing the intrinsic resilience of the model itself, utilizing adversarial training or robustness-oriented loss functions to regularize the graph encoder during the learning phase. These two complementary strategies represent the primary defensive axes currently employed to safeguard graph neural networks against structural perturbations.

Despite their success, these methods mainly rely on the training phase to achieve robustness, which introduces two significant limitations for graph foundation models: (1) *Static Defense Vulnerability*: these foundation models are often deployed in zero-shot scenarios where target-graph adversarial patterns may differ from those observed during training, leading to static defense failure. (2) *Re-Training Cost*: the scale of graph foundation models makes retraining or fine-tuning for specific adversarial attacks computationally prohibitive, rendering traditional train-then-defend paradigms impractical for real-time inference.

In this paper, we propose a novel framework termed TRACE (Test-time gRaph recAlibration for enhancing robust zero-shot inferenCE). Unlike previous methods that treat defense as a static pre-processing or training-time task, TRACE shifts the focus to inference. The key insight is that adversarial attacks disrupt the intrinsic structural-semantic alignment of graphs, which can be diagnosed and neutralized on-the-fly. TRACE introduces a smoothed spectral entropy metric to detect adversarial ego-graphs without ground-truth labels. Upon detection, TRACE applies uniformity-guided anti-attack optimization. Using semantic anchors from pretrained text encoders, TRACE recalibrates a purified graph structure via a gradient-based edge flipping strategy compatible with modern sparse encoders. An empirical case study in Figure 1 illustrates this phenomenon.

The contributions of this work are summarized as follows: (1) *New Problem.* We study realistic yet underexplored test-time robust zero-shot graph inference for graph foundation models, which enhances robustness against adversarial attacks without model retraining. (2) *Novel Methodology.* We propose TRACE, a test-time recalibration framework that combines a non-parametric diagnostic mechanism based on smoothed spectral entropy to discern adversarial from clean graphs with a uniformity-guided optimization objective. It also introduces a first-order gradient-based edge-flipping strategy for modern sparse graph encoders to reconstruct clean graphs in the discrete domain. (3) *Superior Performance.* Extensive experiments across various datasets and adversarial attacks demonstrate that TRACE outperforms state-of-the-art baselines in maintaining high zero-shot accuracy under noisy conditions.

2 Related Work

Graph Adversarial Attack. Graph learning models exhibit significant vulnerability to structural perturbations, as topological connections are frequently more susceptible to covert manipulation than node attributes. Nettack [29] pioneered this field by introducing targeted attacks designed to mislead predictions for specific nodes. Subsequently, Metattack [30] proposed a meta-learning framework for non-targeted attacks to degrade overall model performance. Other approaches, such as Minmax and PGD [31], employ projected gradient descent to maximize supervised loss for structural

modification. CLGA [32] further developed unsupervised attacks using contrastive loss, where edges are flipped based on gradient magnitude. To address challenges in scalability, PRBCD [33] introduced a sparsity-aware optimization attack tailored for large-scale graphs. However, the majority of existing methods operate on dense adjacency matrices or rely on continuous relaxations of the topological space. These characteristics render such methods incompatible with the discrete, sparse edge sets required by modern graph encoders and hinder their integration into graph foundation models.

Graph Adversarial Defense. Existing defense mechanisms typically focus on either enhancing model robustness through adversarial training [34, 35, 36, 26, 27] or eliminating structural noise directly from the graph [37, 38, 39, 23, 22]. For instance, SP-AGCL [35] updates adversarial graphs during the training phase for contrastive optimization, whereas AMD-GCL [27] formulates a min-max problem to maximize augmentation discrepancy for updating adversarial graphs. Regarding purification, LLM4RGNN [22] utilizes fine-tuned LLMs to identify malicious edges and restore missing connections. SG-GSR [40] extracts clean sub-graphs and applies specialized augmentation to preserve structural information. Additionally, DiffSP [23] employs graph diffusion models for prior-free purification, combining non-isotropic forward diffusion with entropy-guided denoising to reconstruct clean distributions. Despite these advancements, adversarial training remains computationally demanding and often fails to generalize to novel attack patterns. Similarly, purification methods are limited by their reliance on known noise distributions and may inadvertently remove intrinsic structural patterns. Crucially, these mechanisms generally lack the zero-shot adaptability required for dynamic test-time defense.

Zero-Shot Graph Learning. In contrast to traditional models trained for task-specific objectives, graph foundation models are pre-trained on large-scale data and adapted to various downstream applications. GraphGPT [41], for example, employs a lightweight projector and instruction tuning to align graph structures with LLMs. InstructGLM [42] reformulates node and link prediction as natural language tasks, thereby removing the requirement for GNN encoders. PromptGFM [43] utilizes prompts to emulate message passing, establishing a unified vocabulary for cross-graph tuning. GFT [44] identifies computational trees as transferable patterns within a shared vocabulary. To eliminate the need for manual labeling, ZeroG [45] leverages LLMs to unify feature and label spaces, while GraphCLIP [25] aligns node attributes with textual summaries via contrastive pre-training. Although these foundation models demonstrate strong generalization, they remain susceptible to structural noise due to the lack of explicit mechanisms for test-time topological recalibration.

3 Preliminaries & Problem Definition

Text-Attributed Graph. Given the entire graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, $\mathcal{V}$ denotes the set of N nodes, $\mathcal{E}$ represents the set of M edges, and $\mathcal{T} = \{T_1, T_2, \cdots, T_N\}$ is the set of N textual descriptions. For instance, in a co-purchasing network, each node represents a product, the attached attribute is the product description, and each edge indicates joint purchases. The topological structure is defined by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where $\mathbf{A}_{ij} = 1$ if nodes i and j are connected, and $\mathbf{A}_{ij} = 0$ otherwise. The node feature matrix $\mathbf{X} \in \mathbb{R}^{N \times F}$ is typically encoded from raw text utilizing modern language models. We address the node classification task over a set of class labels $\mathcal{Y} = \{y_1, y_2, \ldots, y_C\}$, where C indicates the number of classes. Each node $v_i \in \mathcal{V}$ is associated with a ground-truth label $y_i \in \mathcal{Y}$. For the zero-shot classification task, each node i serves as an anchor to sample an ego-graph $G^i = (\mathbf{A}^i, \mathbf{X}^i)$. Consequently, the class of the ego-graph G^i is jointly determined by the structural neighborhood and the corresponding attributes. Here, $\mathbf{A}^i$ and $\mathbf{X}^i$ represent the adjacency matrix and the node feature matrix of the ego-graph centered at node i, respectively.

Graph Adversarial Attack. Attacks on graphs perturb the topological structure to mislead model predictions. We focus on topological attacks, as they are more practical than feature-based attacks in high-privacy scenarios. Formally, given a clean ego-graph $G^i = (\mathbf{A}^i, \mathbf{X}^i)$, an attacker seeks to generate an adversarial graph $\widetilde{G}^i = \arg\min_{G' \in \mathcal{B}(G^i)} p_{c^*}(G')$ that minimizes the probability of the true class c^*. Here, $\mathcal{B}(G^i) = \{G' \mid \|\mathbf{A}' - \mathbf{A}^i\|_0 \leq \Delta\}$ defines the allowable perturbation space, restricting structural modifications to a maximum budget Δ to ensure the attack remains stealthy.

Zero-Shot Inference. We develop our method for the node classification task. Specifically, we utilize the reported GraphCLIP [25] checkpoint as the baseline foundation model, which is pretrained on 200 million tokens across cross-domain large-scale graphs. Let $f_\theta(\cdot)$ denote the graph encoder with weights θ and $g_\phi(\cdot)$ represent the text encoder with weights ϕ. Given an ego-graph G^i and a textual description T_c for the c-th category (e.g., "a paper of [CLASS]"), the model computes a graph

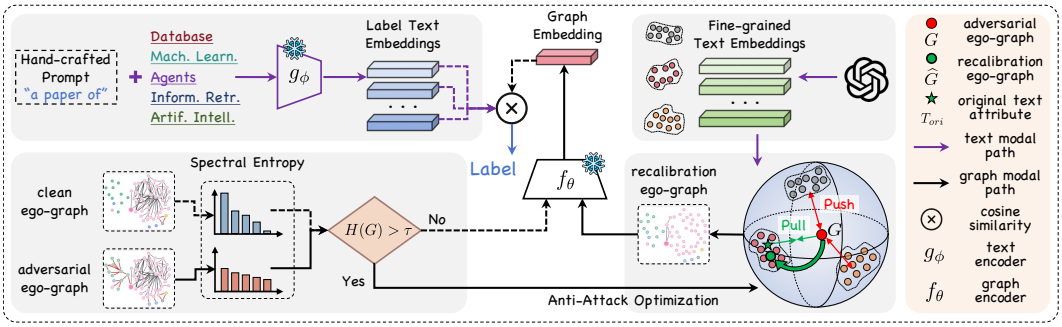

Figure 2: **Overview of the TRACE framework**, which includes three components: spectral entropy diagnosis, fine-grained text generation, and anti-attack optimization. It separates adversarial from clean ego-graphs through the spectral entropy. For adversarial ego-graphs, LLM-generated fine-grained texts serve as the real uniform distribution on a hypersphere. Then, anti-attack optimization pulls the adversarial ego-graph G toward its original text attribute T_{ori} and pushes it away from negative clusters, yielding a recalibrated ego-graph $\widehat{G}$. Finally, the graph embedding is aligned with label text embeddings via cosine similarity for zero-shot classification.

representation $f_\theta(G^i)$ and a text representation $g_\phi(T_c)$. The probability that G^i belongs to category c is determined by the cosine similarity between the graph and text representations:

$$p_c(G^i) = \frac{\exp\left(\cos\left(f_\theta\left(G^i\right), g_\phi\left(T_c\right)\right)\right)}{\sum_{j=1}^{C} \exp\left(\cos\left(f_\theta\left(G^i\right), g_\phi\left(T_j\right)\right)\right)}, \tag{1}$$

where $p_c(G^i)$ is the predicted probability of the c-th class for the input ego-graph G^i. The functions $\exp(\cdot)$ and $\cos(\cdot,\cdot)$ denote the exponential and cosine similarity functions, respectively. For notational simplicity, we omit the superscript i henceforth unless otherwise specified.

Test-Time Graph Recalibration. To bypass expensive adversarial retraining, we formalize *test-time graph recalibration* as a zero-shot reverse-optimization process during inference. Given a perturbed ego-graph $\widetilde{G}$ and frozen encoders, it dynamically rewires the topology to maximize unsupervised alignment with textual semantic priors. This alignment approximates the clean manifold, effectively pruning adversarial edges to ensure robust predictions without ground-truth labels.

4 The Proposed Framework: TRACE

Framework Overview. This paper studies robust zero-shot inference under adversarial attacks. We employ a selective recalibration strategy to neutralize structural noise during inference. To distinguish adversarial graphs from clean ones, normalized smoothed spectral entropy serves as a non-parametric diagnostic trigger. To overcome the lack of test-time ground-truth labels, we formulate a uniformity-guided optimization objective leveraging semantic anchors from pretrained text encoders to recover clean manifold embeddings. To address discrete graph structures and sparse encoder efficiency, we implement a first-order gradient-based edge flipping strategy to reconstruct the clean topology. Figure 2 illustrates the framework, with each component elaborated in subsequent sections.

4.1 Selective Recalibration via Smoothed Spectral Entropy

In real-world scenarios, defenders often lack prior knowledge regarding specific adversarial attack types or the perturbation status of input samples. The indiscriminate application of defense mechanisms risks degrading the generalization of the model on clean data. To address this, we propose normalized smoothed spectral entropy as a non-parametric diagnostic metric to quantify the semantic-topological alignment of ego-graphs.

Metric Formulation. Consider an ego-graph $G = (\mathbf{A}, \mathbf{X})$. To evaluate the alignment between the topological structure $\mathbf{A}$ and the attributes $\mathbf{X}$, we first construct the symmetrically normalized adjacency matrix with self-loops as follows:

$$\tilde{\mathbf{A}} = \tilde{\mathbf{D}}^{-1/2}(\mathbf{A} + \mathbf{I})\tilde{\mathbf{D}}^{-1/2}, \tag{2}$$

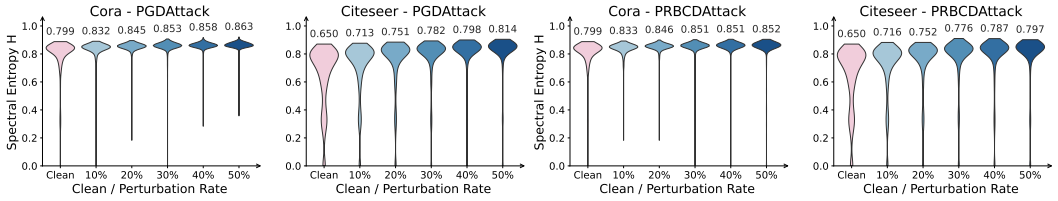

Figure 3: Quantitative analysis of spectral entropy H under PGD [31] and PRBCD [33] attacks. Violin plots visualize the distribution of H across varying perturbation rates, i.e., the greater the attack, the greater the mean entropy. The transition from broad, low-entropy distributions to concentrated, high-entropy distributions serves as a diagnostic basis for identifying adversarial graphs.

where $\mathbf{I}$ denotes the identity matrix and $\tilde{\mathbf{D}}$ represents the degree matrix of $\mathbf{A} + \mathbf{I}$. Subsequently, we apply a first-order Laplacian smoothing operation to obtain the smoothed feature matrix:

$$\mathbf{M} = \tilde{\mathbf{A}}\mathbf{X}. \tag{3}$$

In the context of graph signal processing, this operation functions as a low-pass filter. In clean ego-graphs, adjacent nodes typically possess similar attributes, which results in a highly correlated matrix $\mathbf{M}$ characterized by a rapidly decaying singular value spectrum. Conversely, topological attacks introduce heterophilous edges that force discordant features to aggregate, thereby generating high-frequency spatial noise that flattens the spectrum. We perform Singular Value Decomposition (SVD) [46] on $\mathbf{M}$ to obtain the set of singular values $\{\sigma_1, \sigma_2, \ldots, \sigma_k\}$. By defining the spectral energy distribution as $p_j = \sigma_j / \sum_{m=1}^{k} \sigma_m$, we compute the normalized entropy:

$$H(G) = \frac{-\sum_{j=1}^{k} p_j \log p_j}{\log(k)}. \tag{4}$$

The normalization by $\log(k)$ ensures that $H(G) \in [0, 1]$ remains invariant to the size of the subgraph, providing a standardized measure of disorder where a value of 1.0 signifies maximum structural-semantic conflict.

Empirical Analysis. As illustrated in Figure 3, the evolution of H reveals two consistent spectral signatures. First, a *monotonic mean shift* occurs: as the perturbation rate increases, the mean of H rises steadily (e.g., from 0.650 to 0.814 on CiteSeer under PGD), confirming that attacks systematically degrade the low-rank structure of the feature space. Second, we identify *variance collapse*. While natural ego-graphs exhibit a wide spectrum of structural complexities, adversarial attacks force these diverse topologies into a concentrated, high-entropy regime. This effect, evidenced by the shrinking variance and narrowing of the violin plots, signifies that adversarial perturbations suppress the inherent structural variety of the graph, pushing samples toward a uniform state of maximum disorder. Such a spectral phenomenon allows us to trigger recalibration only when $H(G)$ deviates from the natural entropy distribution.

Practical Application. Building on these observations, we execute a threshold-based trigger to selectively activate recalibration. As diagnosis occurs on individual ego-graphs, we define a threshold $\tau \in [0, 1]$ to distinguish adversarial from clean samples. Optimization is initiated if and only if the normalized spectral entropy exceeds this limit, $H(G) > \tau$. This non-parametric, training-free strategy allows users to calibrate defense intensity based on expected environmental noise. Bypassing optimization for samples where $H(G) \leq \tau$ preserves intrinsic structural priors of clean data.

4.2 Uniformity-Guided Anti-Attack Optimization Objective

After identifying adversarial ego-graphs, the challenge lies in determining a trajectory for adversarial purification within the latent manifold. We formalize a test-time recalibration objective rooted in the principle of uniformity [47]. In contrastive learning, uniformity encourages feature embeddings to be distributed maximally across the unit hypersphere, preserving information entropy and preventing latent space collapse into a degenerate manifold. In our framework, this property acts as a critical geometric constraint, ensuring adversarial samples maintain an intrinsic class-consistent distribution as they decouple from adversarial regions.

A bottleneck lies in the scarcity of available test samples, which prevents accurate estimation of the underlying data distribution and may confuse the optimization direction toward the clean manifold. To

address this, we introduce a geometric prior that leverages the zero-shot capabilities of LLMs. For a dataset comprising C classes, we synthesize M fine-grained textual descriptions per class, generating $M \times C$ text embeddings via a frozen text encoder g_ϕ. These embeddings form class-specific clusters that collectively approximate the uniform distribution of clean data on the hypersphere.

Formally, let the input noise ego-graph be $G = (\mathbf{A} + \delta_{atk}, \mathbf{X})$ with the perturbation δ_{atk}. We treat G as an adversarial anchor and learn an anti-attack perturbation δ_{anti} to recalibrate a purified graph $\widehat{G} = (\mathbf{A} + \delta_{atk} + \delta_{anti}, \mathbf{X})$. The optimization objective is formulated as:

$$\delta^*_{anti} = \arg\max_{\delta_{anti}} \mathcal{L} = \underbrace{-\cos\left(f_\theta(\widehat{G}), f_\theta\left(G\right)\right)}_{\text{Adversarial Decoupling}} + \underbrace{\cos\left(f_\theta(\widehat{G}), g_\phi\left(T_{ori}\right)\right)}_{\text{Semantic Anchor}} \\ \underbrace{-\log \sum_{c=1}^{C} \sum_{m=1}^{M} \exp\left(\cos\left(f_\theta(\widehat{G}), g_\phi(T_c^m)\right)\right)}_{\text{Uniformity-based Guidance}} \quad \text{subject to} \quad \|\delta_{anti}\|_0 \leq \Delta, \tag{5}$$

where three terms guide the recalibration trajectory. The first term forces the optimized topology to diverge from the adversarial anchor, while the second and third terms maintain semantic consistency with the original attributes and promote a uniform distribution across the synthesized class-specific clusters. This joint optimization essentially establishes structure-aware contrastive learning. By balancing positive semantic alignment with negative uniformity regularization, the latter two terms map to the InfoNCE objective, an equivalence we formally establish in Proposition 1.

Proposition 1. *Maximizing the optimization objective $\mathcal{L}$ is equivalent to simultaneously executing structural deviation from the adversarial input graph and performing structure-aware contrastive learning to preserve semantic consistency.*

Proof Sketch. The overall objective decomposes into two components. The first term minimizes the cosine similarity between the purified graph and the adversarial input, driving the optimized topology to diverge from the adversarial perturbations. By defining the purified embedding as the query, the original text embedding as the positive key, and the synthesized text embeddings as the negative pool, the sum of the second and third terms reduces to a negative InfoNCE loss. Consequently, maximizing this joint objective identifies a structurally divergent topology while preserving semantic consistency through contrastive alignment. The complete proof is provided in Appendix A.2. □

4.3 Anti-Attack Optimization for Structural Recalibration

Optimizing Eq. (5) is challenging given the discrete, non-differentiable adjacency matrix $\mathbf{A}$. While traditional adversarial attacks [35, 48] optimize continuous relaxations of dense matrices, this $\mathcal{O}(N^2)$ paradigm is computationally prohibitive and incompatible with sparse graph encoders. Moreover, brute-force search for discrete sparse perturbations within budget Δ is NP-hard. To overcome dense matrix bottlenecks, we adopt a first-order gradient-based structural modification strategy. Acting as an anti-attack model, this optimization uses adversarial generation to efficiently solve for the optimal anti-attack perturbation δ^*_{anti} and recalibrate the purified graph $\widehat{G}$ in the sparse domain.

Continuous Gradient Approximation. Because modern sparse graph encoders lack computational pathways to backpropagate gradients through non-existent edges, direct differentiation of the topological space is infeasible. We approximate structural gradients by differentiating the anti-attack recalibration objective $\mathcal{L}$ in Eq. (5) with respect to the continuous node feature matrix $\mathbf{X}$. Computing gradients for input features yields the feature gradient matrix $\nabla_{\mathbf{X}}\mathcal{L} \in \mathbb{R}^{N\times F}$, circumventing the need to instantiate dense topological matrices.

Edge Importance Scoring. Within the message-passing of graph encoders, nodes update representations by aggregating attributes from topological neighbors. Hence, the impact of topological alterations on the overall objective can be estimated via a first-order Taylor expansion. Specifically, the structural gradient is approximated as the inner product between the target node's feature gradient and the neighbor's original feature:

$$\mathbf{S}_{ij} = \left(\nabla_{\mathbf{x}_i}\mathcal{L}\right)^\top \mathbf{x}_j, \tag{6}$$

where $\mathbf{S} = (\nabla_{\mathbf{X}}\mathcal{L})\mathbf{X}^\top \in \mathbb{R}^{N\times N}$ serves as the edge importance scoring matrix for the localized ego-graph. Based on directional gradients, a pronounced positive score $\mathbf{S}_{ij}$ at a non-existent edge

indicates that inserting a connection there provides the steepest ascent to maximize the anti-attack objective $\mathcal{L}$. Conversely, a highly negative score at an existing edge signifies that removing this connection also maximizes the anti-attack objective $\mathcal{L}$.

Guided by $\mathbf{S}$, we rewire the ego-graph under edge addition and deletion budgets (Δ_{add} and Δ_{drop}), each proportional to the number of existing edges. By applying mutually exclusive masks to existing and non-existing edges, we add the Δ_{add} non-edge candidates with the highest positive scores and remove the Δ_{drop} existing edges with the most negative scores.

Theoretical Intuition of Edge Scoring. To maximize the anti-attack objective $\mathcal{L}$ through discrete topological modifications, we analyze the impact of edge flips via a first-order Taylor expansion. In message-passing, a node's hidden representation is approximately a linear aggregation of neighbors' features, e.g., $\mathbf{h}_i \approx \sum_k \mathbf{A}_{ik}\mathbf{x}_k$. Thus, the objective function variation $\Delta\mathcal{L}$ caused by a discrete edge flip $\Delta\mathbf{A}_{ij} \in \{-1, 1\}$ is bounded by the exact structural gradient:

$$\Delta\mathcal{L} \approx \frac{\partial\mathcal{L}}{\partial\mathbf{A}_{ij}}\Delta\mathbf{A}_{ij} = \left((\nabla_{\mathbf{h}_i}\mathcal{L})^\top \mathbf{x}_j\right)\Delta\mathbf{A}_{ij}. \tag{7}$$

However, modern sparse graph encoders lack computational pathways to backpropagate latent gradients $\nabla_{\mathbf{h}_i}\mathcal{L}$ through non-existent edges (i.e., when $\mathbf{A}_{ij} = 0$). To circumvent the $\mathcal{O}(N^2)$ memory bottleneck of dense continuous relaxation, we substitute the inaccessible latent gradient with the input feature gradient $\nabla_{\mathbf{x}_i}\mathcal{L}$ as a continuous surrogate:

$$\mathbf{S}_{ij} = (\nabla_{\mathbf{x}_i}\mathcal{L})^\top \mathbf{x}_j. \tag{8}$$

Geometrically, this score estimates the projection of neighbor feature $\mathbf{x}_j$ onto the steepest ascent direction of target node i. Guided by this proxy, inserting non-edges with highly positive scores ($\Delta\mathbf{A}_{ij} = +1$) or dropping existing edges with highly negative scores ($\Delta\mathbf{A}_{ij} = -1$) mathematically guarantees $\Delta\mathcal{L} > 0$, maximizing the recalibration objective directly within the sparse domain. For formal proofs and detailed discussions on sparse autograd limitations, see Appendix A.3.

5 Experiment

5.1 Experimental Settings

Datasets. To evaluate the effectiveness of TRACE, we conduct experiments across diverse graph benchmarks. Given that GraphCLIP checkpoint[2] was pre-trained on ogbn-arxiv [49], Arxiv-2023 [50], PubMed [51], ogbn-products [52], and Reddit [53], we evaluate our framework on ten datasets to validate the zero-shot robustness. Specifically, we evaluate our framework's robustness across two benchmark sets. First, we select five widely-adopted benchmarks: Cora [54], CiteSeer [1], WikiCS [55], History [56], and Children [56]. Second, to further assess in extreme scenarios, we evaluate on five highly heterophilous graphs: Texas [57], Cornell [57], Washington [58], Wisconsin [57], and Actor [57]. Detailed statistics and descriptions for these datasets are provided in Appendix C.

Baselines. To evaluate performance, we benchmark TRACE against a diverse suite of competitive models, including three self-supervised learning methods (DGI [59], BGRL [60], and G2P2 [61]) and four graph foundation models (GraphGPT [41], OFA [62], ZeroG [45], and GraphCLIP [25]). The zero-shot capabilities of self-supervised and discriminative approaches are quantified by computing the cosine similarity between node embeddings and text label embeddings. Conversely, for generative models, we rely on their inherent text-generation mechanisms to produce zero-shot predictions. A comprehensive description of these baselines is provided in Appendix D.

Implementation Details[3]. All experiments use a single NVIDIA RTX 3090 GPU. We evaluate performance under PGD [31] and PRBCD [33] attacks across varying perturbation rates. Following settings in [25], we employ the pre-trained GraphGPS model [63] as graph encoder f_θ and all-MiniLM-L6-v2[4] [55] as text encoder g_ϕ. We synthesize $M = 100$ textual descriptions per class. The entropy diagnostic threshold τ is set to 0.7. For budget-constrained edge flipping, both addition Δ_{add} and deletion Δ_{drop} ratios are constrained to 30% of existing edges within the localized ego-graph. We set the number of steps for anti-attacks as $E = 10$. Further details are provided in Appendix B.

[2] https://github.com/ZhuYun97/GraphCLIP

[3] Our code is available at https://github.com/CCChen-GEEX/TRACE.

[4] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Table 1: Zero-shot robust accuracies on five graph datasets under attacks in terms of accuracy (in %).

	Methods		PGDAttack				PRBCDAttack				
Data	Ptb. Rate	Clean	10%	20%	30%	40%	10%	20%	30%	40%	AVG
Cora	DGI	28.7$_{\pm 0.7}$	28.7$_{\pm 0.7}$	27.9$_{\pm 0.5}$	27.4$_{\pm 0.8}$	27.1$_{\pm 1.1}$	27.5$_{\pm 1.0}$	26.8$_{\pm 0.8}$	25.2$_{\pm 1.0}$	24.1$_{\pm 0.8}$	26.8
	BGRL	24.8$_{\pm 1.1}$	23.7$_{\pm 2.0}$	23.2$_{\pm 1.8}$	22.8$_{\pm 2.4}$	21.8$_{\pm 1.8}$	23.8$_{\pm 2.0}$	23.6$_{\pm 2.1}$	22.2$_{\pm 2.3}$	20.9$_{\pm 1.8}$	22.8
	G2P2	61.1$_{\pm 2.0}$	57.5$_{\pm 1.9}$	51.9$_{\pm 1.1}$	49.4$_{\pm 1.9}$	46.1$_{\pm 1.2}$	51.6$_{\pm 1.3}$	43.3$_{\pm 1.0}$	34.5$_{\pm 0.8}$	28.6$_{\pm 1.1}$	45.4
	GraphGPT	25.5$_{\pm 1.2}$	23.5$_{\pm 1.4}$	21.2$_{\pm 1.5}$	19.0$_{\pm 1.6}$	17.5$_{\pm 1.8}$	21.8$_{\pm 1.5}$	18.5$_{\pm 1.7}$	16.2$_{\pm 1.9}$	14.8$_{\pm 2.0}$	19.1
	OFA	37.3$_{\pm 1.4}$	35.1$_{\pm 1.5}$	32.8$_{\pm 1.7}$	30.5$_{\pm 1.8}$	28.2$_{\pm 1.9}$	33.5$_{\pm 1.6}$	30.2$_{\pm 1.9}$	27.8$_{\pm 2.1}$	25.4$_{\pm 2.3}$	30.4
	ZeroG	65.3$_{\pm 1.6}$	63.3$_{\pm 1.8}$	60.7$_{\pm 1.9}$	60.1$_{\pm 1.7}$	58.8$_{\pm 1.8}$	61.4$_{\pm 1.8}$	57.8$_{\pm 2.0}$	55.8$_{\pm 2.1}$	53.5$_{\pm 2.2}$	58.9
	GraphCLIP	66.2$_{\pm 2.0}$	63.6$_{\pm 1.3}$	61.3$_{\pm 1.9}$	59.6$_{\pm 1.8}$	59.3$_{\pm 1.4}$	60.0$_{\pm 1.4}$	56.7$_{\pm 2.5}$	54.6$_{\pm 1.7}$	53.3$_{\pm 2.1}$	58.6
	TRACE	**66.4**$_{\pm 2.2}$	**64.8**$_{\pm 1.5}$	**61.9**$_{\pm 1.4}$	**61.1**$_{\pm 1.6}$	**60.1**$_{\pm 1.5}$	**62.7**$_{\pm 1.8}$	**59.6**$_{\pm 2.1}$	**58.0**$_{\pm 1.5}$	**56.6**$_{\pm 1.8}$	**60.6**
CiteSeer	DGI	22.5$_{\pm 1.1}$	22.1$_{\pm 1.0}$	21.5$_{\pm 0.5}$	21.3$_{\pm 1.1}$	20.4$_{\pm 1.5}$	21.8$_{\pm 1.3}$	20.6$_{\pm 0.5}$	19.3$_{\pm 0.9}$	18.7$_{\pm 0.6}$	20.7
	BGRL	25.3$_{\pm 1.5}$	25.1$_{\pm 1.6}$	24.4$_{\pm 1.8}$	23.1$_{\pm 1.6}$	22.8$_{\pm 1.1}$	23.7$_{\pm 1.4}$	22.9$_{\pm 1.6}$	21.9$_{\pm 1.9}$	20.4$_{\pm 2.1}$	23.0
	G2P2	49.2$_{\pm 0.9}$	45.7$_{\pm 1.4}$	41.0$_{\pm 1.5}$	37.5$_{\pm 0.7}$	34.7$_{\pm 1.5}$	40.5$_{\pm 1.7}$	33.2$_{\pm 1.0}$	26.6$_{\pm 1.3}$	24.3$_{\pm 0.8}$	35.4
	GraphGPT	14.0$_{\pm 1.5}$	12.8$_{\pm 1.6}$	11.5$_{\pm 1.7}$	10.2$_{\pm 1.8}$	9.0$_{\pm 1.8}$	11.5$_{\pm 1.7}$	9.8$_{\pm 1.8}$	8.5$_{\pm 1.9}$	7.8$_{\pm 2.0}$	10.1
	OFA	40.5$_{\pm 1.2}$	38.0$_{\pm 1.4}$	35.5$_{\pm 1.6}$	33.2$_{\pm 1.5}$	31.0$_{\pm 1.7}$	35.8$_{\pm 1.5}$	32.5$_{\pm 1.8}$	30.1$_{\pm 1.9}$	28.5$_{\pm 2.0}$	33.1
	ZeroG	51.6$_{\pm 1.2}$	49.5$_{\pm 1.3}$	47.8$_{\pm 1.5}$	46.5$_{\pm 1.4}$	45.1$_{\pm 1.5}$	47.5$_{\pm 1.4}$	44.8$_{\pm 1.6}$	43.1$_{\pm 1.7}$	42.0$_{\pm 1.8}$	45.8
	GraphCLIP	65.7$_{\pm 1.5}$	61.0$_{\pm 1.1}$	58.9$_{\pm 1.3}$	56.2$_{\pm 1.2}$	55.6$_{\pm 1.8}$	56.8$_{\pm 1.4}$	52.5$_{\pm 1.0}$	49.6$_{\pm 2.2}$	48.2$_{\pm 1.6}$	54.9
	TRACE	**66.4**$_{\pm 1.6}$	**63.6**$_{\pm 1.2}$	**61.8**$_{\pm 1.7}$	**61.4**$_{\pm 1.6}$	**60.1**$_{\pm 1.2}$	**61.0**$_{\pm 1.0}$	**59.4**$_{\pm 1.4}$	**57.8**$_{\pm 1.6}$	**56.9**$_{\pm 1.6}$	**60.3**
WikiCS	DGI	20.7$_{\pm 0.1}$	19.8$_{\pm 0.2}$	19.2$_{\pm 0.3}$	18.8$_{\pm 0.3}$	18.1$_{\pm 0.1}$	19.1$_{\pm 0.5}$	18.5$_{\pm 0.6}$	18.5$_{\pm 0.8}$	17.3$_{\pm 0.7}$	18.7
	BGRL	19.8$_{\pm 0.1}$	19.0$_{\pm 0.0}$	18.9$_{\pm 0.1}$	18.5$_{\pm 0.1}$	18.0$_{\pm 0.2}$	19.0$_{\pm 0.4}$	18.2$_{\pm 0.5}$	17.6$_{\pm 0.5}$	16.9$_{\pm 0.4}$	18.3
	G2P2	37.0$_{\pm 0.0}$	35.2$_{\pm 0.2}$	34.5$_{\pm 0.3}$	33.7$_{\pm 0.2}$	32.9$_{\pm 0.2}$	35.7$_{\pm 0.2}$	35.0$_{\pm 0.2}$	34.5$_{\pm 0.3}$	33.9$_{\pm 0.2}$	34.4
	GraphGPT	35.6$_{\pm 0.2}$	33.8$_{\pm 0.4}$	32.5$_{\pm 0.5}$	31.0$_{\pm 0.6}$	29.8$_{\pm 0.6}$	30.5$_{\pm 0.4}$	28.2$_{\pm 0.5}$	26.8$_{\pm 0.6}$	25.5$_{\pm 0.7}$	29.8
	OFA	32.8$_{\pm 0.6}$	31.0$_{\pm 0.7}$	29.8$_{\pm 0.8}$	28.5$_{\pm 0.9}$	27.2$_{\pm 0.9}$	28.2$_{\pm 0.8}$	25.8$_{\pm 1.0}$	24.5$_{\pm 1.1}$	23.0$_{\pm 1.2}$	27.3
	ZeroG	47.9$_{\pm 0.7}$	46.0$_{\pm 0.8}$	45.1$_{\pm 0.9}$	44.2$_{\pm 1.0}$	43.5$_{\pm 1.0}$	42.1$_{\pm 0.9}$	39.5$_{\pm 1.1}$	38.0$_{\pm 1.2}$	36.5$_{\pm 1.3}$	41.9
	GraphCLIP	70.8$_{\pm 0.0}$	66.6$_{\pm 0.2}$	64.7$_{\pm 0.5}$	63.5$_{\pm 0.6}$	62.7$_{\pm 0.4}$	60.0$_{\pm 0.9}$	57.0$_{\pm 1.6}$	56.0$_{\pm 1.4}$	54.4$_{\pm 1.5}$	60.6
	TRACE	**71.0**$_{\pm 0.0}$	**68.3**$_{\pm 0.4}$	**66.9**$_{\pm 0.8}$	**65.7**$_{\pm 0.9}$	**64.9**$_{\pm 0.6}$	**62.7**$_{\pm 1.0}$	**59.5**$_{\pm 1.0}$	**57.6**$_{\pm 1.6}$	**56.0**$_{\pm 1.3}$	**62.7**
History	DGI	19.8$_{\pm 0.3}$	19.1$_{\pm 0.2}$	18.2$_{\pm 0.4}$	17.6$_{\pm 0.5}$	16.4$_{\pm 0.4}$	19.3$_{\pm 0.4}$	18.6$_{\pm 0.4}$	17.6$_{\pm 0.5}$	16.8$_{\pm 0.3}$	18.0
	BGRL	17.7$_{\pm 0.2}$	16.4$_{\pm 0.3}$	15.9$_{\pm 0.2}$	15.4$_{\pm 0.2}$	14.9$_{\pm 0.4}$	16.2$_{\pm 0.2}$	15.5$_{\pm 0.6}$	15.3$_{\pm 0.7}$	15.1$_{\pm 0.9}$	15.6
	G2P2	19.7$_{\pm 0.6}$	19.0$_{\pm 0.6}$	18.2$_{\pm 0.7}$	17.6$_{\pm 0.8}$	17.4$_{\pm 0.9}$	17.2$_{\pm 1.3}$	16.3$_{\pm 1.8}$	16.3$_{\pm 2.1}$	15.6$_{\pm 2.0}$	17.2
	GraphGPT	16.4$_{\pm 0.1}$	15.5$_{\pm 0.2}$	14.8$_{\pm 0.3}$	14.2$_{\pm 0.3}$	13.5$_{\pm 0.4}$	12.2$_{\pm 0.3}$	11.2$_{\pm 0.4}$	10.2$_{\pm 0.5}$	9.2$_{\pm 0.5}$	12.6
	OFA	15.3$_{\pm 0.9}$	14.5$_{\pm 1.0}$	13.8$_{\pm 1.1}$	13.2$_{\pm 1.2}$	12.5$_{\pm 1.2}$	11.5$_{\pm 1.1}$	10.5$_{\pm 1.2}$	9.5$_{\pm 1.3}$	8.5$_{\pm 1.4}$	11.8
	ZeroG	39.5$_{\pm 0.7}$	38.3$_{\pm 0.8}$	37.9$_{\pm 0.9}$	37.9$_{\pm 0.9}$	37.1$_{\pm 1.0}$	30.8$_{\pm 0.9}$	28.4$_{\pm 1.0}$	26.8$_{\pm 1.1}$	26.0$_{\pm 1.2}$	32.9
	GraphCLIP	52.3$_{\pm 0.5}$	50.6$_{\pm 0.6}$	49.9$_{\pm 0.5}$	49.8$_{\pm 0.4}$	48.7$_{\pm 0.8}$	39.1$_{\pm 0.6}$	35.8$_{\pm 0.5}$	33.8$_{\pm 0.7}$	32.6$_{\pm 0.6}$	42.5
	TRACE	**54.0**$_{\pm 0.7}$	**52.9**$_{\pm 0.5}$	**52.3**$_{\pm 0.6}$	**52.4**$_{\pm 0.6}$	**51.6**$_{\pm 0.9}$	**43.8**$_{\pm 0.6}$	**41.3**$_{\pm 0.7}$	**39.2**$_{\pm 0.8}$	**37.9**$_{\pm 0.8}$	**46.4**
Children	DGI	22.2$_{\pm 0.3}$	20.4$_{\pm 0.4}$	19.7$_{\pm 0.5}$	19.8$_{\pm 0.6}$	19.8$_{\pm 0.6}$	20.0$_{\pm 0.8}$	18.7$_{\pm 1.8}$	17.9$_{\pm 1.1}$	17.9$_{\pm 1.0}$	19.6
	BGRL	14.9$_{\pm 0.1}$	14.4$_{\pm 0.2}$	14.4$_{\pm 0.2}$	14.1$_{\pm 0.3}$	13.8$_{\pm 0.2}$	12.4$_{\pm 0.5}$	11.4$_{\pm 0.3}$	10.3$_{\pm 0.3}$	10.0$_{\pm 0.3}$	12.9
	G2P2	12.5$_{\pm 0.1}$	12.3$_{\pm 0.1}$	12.1$_{\pm 0.1}$	11.7$_{\pm 0.2}$	11.6$_{\pm 0.2}$	12.4$_{\pm 0.5}$	12.0$_{\pm 0.8}$	11.8$_{\pm 0.9}$	11.6$_{\pm 1.0}$	11.9
	GraphGPT	12.4$_{\pm 0.8}$	12.0$_{\pm 0.9}$	11.5$_{\pm 1.0}$	11.0$_{\pm 1.1}$	10.5$_{\pm 1.1}$	10.2$_{\pm 1.0}$	9.5$_{\pm 1.1}$	9.0$_{\pm 1.2}$	8.5$_{\pm 1.3}$	10.3
	OFA	5.9$_{\pm 0.4}$	5.5$_{\pm 0.5}$	5.1$_{\pm 0.6}$	4.8$_{\pm 0.6}$	4.5$_{\pm 0.7}$	4.8$_{\pm 0.6}$	4.4$_{\pm 0.7}$	4.0$_{\pm 0.8}$	3.5$_{\pm 0.9}$	4.6
	ZeroG	15.7$_{\pm 0.4}$	15.5$_{\pm 0.5}$	15.0$_{\pm 0.6}$	14.8$_{\pm 0.7}$	14.5$_{\pm 0.7}$	13.5$_{\pm 0.6}$	12.8$_{\pm 0.7}$	12.5$_{\pm 0.8}$	12.0$_{\pm 0.9}$	13.8
	GraphCLIP	23.1$_{\pm 0.0}$	21.0$_{\pm 0.2}$	20.0$_{\pm 0.3}$	19.7$_{\pm 0.3}$	19.3$_{\pm 0.3}$	18.2$_{\pm 0.2}$	18.1$_{\pm 0.3}$	17.9$_{\pm 0.3}$	16.9$_{\pm 0.3}$	18.9
	TRACE	**23.3**$_{\pm 0.3}$	**23.1**$_{\pm 0.4}$	**22.5**$_{\pm 0.3}$	**22.1**$_{\pm 0.4}$	**21.7**$_{\pm 0.4}$	**20.1**$_{\pm 0.1}$	**19.2**$_{\pm 0.2}$	**18.8**$_{\pm 0.3}$	**18.2**$_{\pm 0.3}$	**20.7**

5.2 Experimental Results

Comparison on Graphs with Adversarial Attacks. *Firstly*, Table 1 proves that TRACE achieves the highest average accuracy across all datasets under both PGD and PRBCD attacks. Notably, our approach bypasses unnecessary optimization on clean samples, preventing over-purification and preserving intrinsic structural priors. *Furthermore*, the robustness of TRACE becomes evident as noise boosts. At a 40% perturbation budget, self-supervised methods, such as G2P2, and foundation models, such as GraphCLIP, undergo performance degradation. Under a 40% PRBCD attack on the History dataset, the accuracy of GraphCLIP drops sharply to 32.6%, whereas TRACE maintains 37.9% accuracy. This margin confirms that the proposed mechanism neutralizes complex topological noise and guides adversarial ego-graphs back toward a clean manifold during test-time inference. *Additionally*, the results on Children highlight the limitations of existing methods in zero-shot settings. Although recent graph foundation models demonstrate better general capabilities, they still lack dynamic adaptability to unseen attacks. TRACE mitigates this vulnerability, achieving 20.7% average accuracy on Children, compared to 18.9% for GraphCLIP. By utilizing textual semantics to guide structural recalibration, the proposed framework provides stable geometric guidance without requiring ground-truth supervision, validating its zero-shot adaptability across highly complex domains.

Comparison on Heterophilous Graphs. Table 2 demonstrates that TRACE consistently achieves SOTA zero-shot performance across all evaluated heterophilous benchmarks. The proposed framework reaches a leading average accuracy of 32.4%, yielding a substantial improvement over GraphCLIP. These empirical results highlight the vulnerability of existing methods; self-supervised models and generative architectures encounter performance degradation on non-homophilous graphs due to their intrinsic reliance on homophily assumptions. In contrast, TRACE successfully mitigates this bias. By utilizing text-attributed semantic anchors rather than relying on local topological proximity, the framework maintains robust adaptability across highly diverse and structurally inverted domains. This uniform superiority confirms that the proposed geometric guidance mechanism provides a reliable optimization trajectory, effectively navigating complex structural-semantic conflicts without requiring dataset-specific fine-tuning.

Table 2: Zero-shot robust accuracies on heterophilous noise graphs in terms of accuracy (in %).

Methods	Texas	Cornell	Wash.	Wisc.	Actor	AVG
DGI	$11.2_{\pm 7.2}$	$14.7_{\pm 5.8}$	$16.9_{\pm 4.0}$	$12.1_{\pm 3.5}$	$29.8_{\pm 0.7}$	16.9
BGRL	$10.9_{\pm 5.9}$	$12.3_{\pm 5.5}$	$12.4_{\pm 4.3}$	$13.4_{\pm 4.1}$	$29.9_{\pm 0.8}$	15.8
G2P2	$14.2_{\pm 4.0}$	$15.9_{\pm 6.5}$	$13.9_{\pm 5.0}$	$21.9_{\pm 2.8}$	$15.9_{\pm 0.5}$	16.4
GraphGPT	$18.5_{\pm 5.3}$	$11.5_{\pm 6.4}$	$11.4_{\pm 3.2}$	$12.9_{\pm 1.5}$	$34.7_{\pm 0.8}$	17.8
OFA	$11.8_{\pm 3.4}$	$18.8_{\pm 4.9}$	$6.0_{\pm 4.1}$	$4.8_{\pm 3.4}$	$31.5_{\pm 0.6}$	14.6
ZeroG	$45.8_{\pm 6.5}$	$10.5_{\pm 5.8}$	$8.3_{\pm 2.5}$	$12.7_{\pm 5.8}$	$40.3_{\pm 0.2}$	23.5
GraphCLIP	$45.8_{\pm 8.4}$	$15.4_{\pm 3.2}$	$13.0_{\pm 3.9}$	$11.3_{\pm 1.2}$	$47.5_{\pm 0.7}$	26.6
TRACE	$\mathbf{50.5}_{\pm 6.5}$	$\mathbf{19.0}_{\pm 4.5}$	$\mathbf{17.4}_{\pm 5.3}$	$\mathbf{26.4}_{\pm 4.1}$	$\mathbf{48.7}_{\pm 0.6}$	**32.4**

Ablation Study. As depicted in Table 3, we conduct comprehensive experiments on different model variants to explore the contributions of each proposed module. TRACE w/o SE represents TRACE without the spectral entropy filtering. TRACE w/o SA denotes TRACE without the semantic anchor term in the optimization objective. TRACE w/o UG indicates TRACE without the uniformity-guided optimization strategy. Finally, TRACE w/o SA&UG signifies the simultaneous removal of both semantic anchors and uniformity guidance. From the results, it is observed that the performance of the full model consistently outperforms all variants across different datasets and perturbation rates. This observation indicates that the spectral entropy filtering effectively identifies adversarial perturbations, while semantic anchors and uniformity guidance jointly facilitate the recovery of the clean manifold. Thus, the effectiveness of each proposed component is sufficiently validated.

Table 3: Ablation studies (in %) of the proposed TRACE in different settings under PRBCD Attack.

Dataset	Cora				CiteSeer				History			
PRBCDAttack Rate	10%	20%	30%	40%	10%	20%	30%	40%	10%	20%	30%	40%
GraphCLIP Baseline	60.0	56.7	54.6	53.3	56.8	52.5	49.6	48.2	39.1	35.8	33.8	32.6
TRACE w/o SE	62.2	59.4	57.8	56.1	60.4	58.7	56.5	55.6	43.3	40.7	38.7	37.0
TRACE w/o SA	61.7	59.2	57.2	54.9	57.8	55.9	54.5	52.9	41.7	38.9	36.6	35.2
TRACE w/o UG	62.5	59.1	57.6	56.4	60.8	58.6	57.0	55.6	43.3	41.0	38.7	37.6
TRACE w/o SA&UG	61.7	58.6	57.3	55.9	57.6	55.1	53.5	53.1	40.4	37.4	35.4	34.2
Full Model	**62.7**	**59.6**	**58.0**	**56.6**	**61.0**	**59.4**	**57.8**	**56.9**	**43.8**	**41.3**	**39.2**	**37.9**

Sensitivity Analysis. Figure 4 (a) investigates the sensitivity to the edge addition (Δ_{add}) and deletion (Δ_{drop}) ratios. Although an exhaustive grid search across all parameter combinations yields the optimal performance for a specific dataset, the heatmaps indicate that constraining both Δ_{add} and Δ_{drop} to 30% consistently enhances robustness. This demonstrates that a balanced and moderate modification budget is sufficient to approximate the clean manifold without dataset-specific hyperparameter tuning. Furthermore, Figure 4 (b) evaluates the impact of the spectral entropy threshold τ, which controls recalibration module activation. The line plots reveal $\tau = 0.7$ as a highly effective operational point. Although applying this threshold introduces a slight performance degradation on unperturbed samples, it consistently and significantly improves classification accuracy across a wide range of attack intensities (10% to 40% perturbation rates). This confirms that $\tau = 0.7$ yields a reliable trade-off, prioritizing critical defense capabilities over severe structural noise while minimizing impact on clean-sample performance.

Visualization. The visualizations in Figure 5 assess the effectiveness of TRACE. Figure 5 (a) illustrates angular density plots of synthesized fine-grained text descriptions mapped onto a hypersphere for Cora and CiteSeer. While the descriptions within each class concentrate on discriminative semantic space, the global distribution across all classes is uniform, which confirms the underlying assumption of the uniformity-guided optimization objective. Figure 5 (b) presents t-SNE visualization of embeddings on CiteSeer under various conditions. The embeddings of the original graph display cluster boundaries that are subsequently disrupted by the PRBCD attack, which leads to entangled

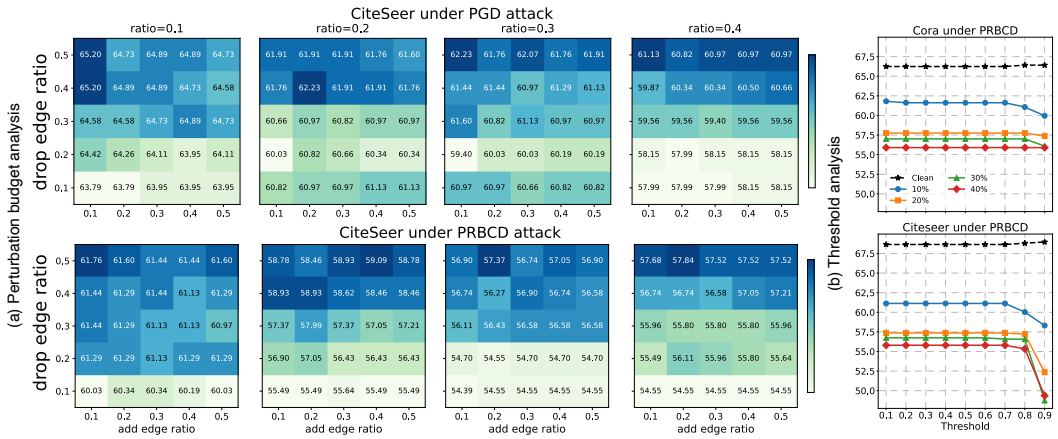

Figure 4: (a) Perturbation budget analysis showing the impact of addition/deletion under varying perturbation rates. (b) Threshold analysis evaluating the effect across different perturbation rates.

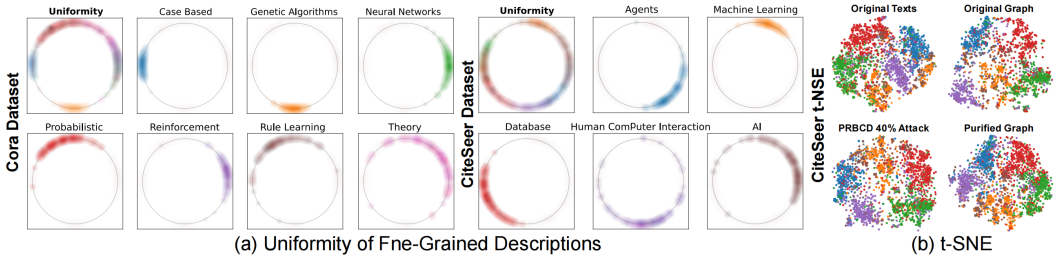

Figure 5: (a) Angular density plots of fine-grained text descriptions proving uniform distribution across the hypersphere. (b) t-SNE visualization of ego-graph embeddings, showing entanglement from PRBCD attack (40% perturbation rate) and successful manifold recovery via the purified graph.

representations. Embeddings derived from the purified graph generated by TRACE recover the cluster structures and align with the original semantic manifold. This evidence validates the capability of the method to neutralize topological perturbations and restore the clean representation space.

6 Conclusion

In this paper, we investigated robust zero-shot inference under adversarial topological attacks. To this end, we proposed TRACE, a novel training-free test-time structural recalibration framework. By introducing normalized smoothed spectral entropy as a non-parametric diagnostic trigger, our method selectively identifies adversarial graphs without degrading clean graph performance. For detected adversarial graphs, we formulated a uniformity-guided optimization objective utilizing semantic anchors from frozen language models to approximate the clean manifold. To overcome computational bottlenecks of sparse autograd and discrete structural search, we designed an efficient continuous feature gradient proxy guiding topological rewiring. Extensive evaluations demonstrate that TRACE effectively neutralizes structural noise and achieves robust zero-shot inference without adversarial retraining. Although our framework is model-agnostic, its generalizability across diverse architectural backbones requires empirical validation. Future research aims to address these challenges by evaluating our method on broader open-source models and expanding our test-time recalibration paradigm to encompass generalized adversarial defense scenarios.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their valuable and constructive comments. This work was supported in part by the National Key R&D Program of China under Grant 2025YFE0202900, in part by Science and Technology Development Fund (FDCT-MOST), Macao SAR under Grant 0007/2025/AMJ, in part by the National Natural Science Foundation of China under Grants 62576253 and 62476109, and in part by the Fundamental Research Funds for the Central Universities.

References

[1] C Lee Giles, Kurt D Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pages 89–98, 1998.

[2] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[3] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

[4] Yunhui Liu, Yue Liu, Yongchao Liu, Tao Zheng, Stan Z Li, Xinwang Liu, and Tieke He. Beyond the academic monoculture: A unified framework and industrial perspective for attributed graph clustering. *arXiv preprint arXiv:2603.20829*, 2026.

[5] Zhongjian Zhang, Mengmei Zhang, Dehua Xu, Rongjun Shi, Jianfeng Liu, Fuli Meng, Huajian Xu, Xiao Wang, Ruijia Wang, Junze Chen, et al. Friskgpt: A generative foundation model for financial risk detection. In *Proceedings of the ACM Web Conference 2026*, pages 7733–7744, 2026.

[6] Yunhui Liu, Xudong Jin, Qizhuo Xie, Chunhui Zhao, Xiao Luo, Tao Zheng, Bin Chong, and Tieke He. Uncovering semantic hierarchies in text-attributed graphs via variational em-based llm-gnn synergy. *arXiv preprint arXiv:2610.24624*, 2026.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[8] Bo Peng, Jie Lu, Guangquan Zhang, and Zhen Fang. An information-theoretical framework for understanding out-of-distribution detection with pretrained vision-language models. *Advances in Neural Information Processing Systems*, 38:1930–1957, 2026.

[9] Bo Peng, Yuanwei Hu, Bo Liu, Ling Chen, Jie Lu, and Zhen Fang. Delving into spectral clustering with vision-language representations. In *The Fourteenth International Conference on Learning Representations*, 2026.

[10] Bo Peng, Jie Lu, Guangquan Zhang, and Zhen Fang. On the provable importance of gradients for autonomous language-assisted image clustering. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19805–19815. IEEE, 2025.

[11] Bo Peng, Jie Lu, Zhen Fang, and Guangquan Zhang. A close look at negative label guided out-of-distribution detection in pre-trained vision-language models. In *Forty-third International Conference on Machine Learning*, 2026.

[12] Bo Peng, Jie Lu, Guangquan Zhang, and Zhen Fang. Debiased negative mining improves out-of-distribution detection with pre-trained vision-language models. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 1171–1181, 2026.

[13] Zhoujun Ye, Yiwei Fu, Qiyun Huang, Jie Yang, Dongjie Wang, and Xiao Luo. Beyond global alignment: Structured compositional reasoning for vision-language models. In *The Fortieth Annual Conference on Neural Information Processing Systems*, 2026.

[14] Jiawei Liu, Cheng Yang, Zhiyuan Lu, Junze Chen, Yibo Li, Mengmei Zhang, Ting Bai, Yuan Fang, Lichao Sun, Philip S Yu, et al. Graph foundation models: Concepts, opportunities and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[15] Zehong Wang, Chuxu Zhang, Jundong Li, Nitesh Chawla, and Yanfang Ye. Graph foundation models: Challenges, methods, and open questions. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6184–6194, 2025.

[16] Shuo Wang, Bokui Wang, Zhixiang Shen, Boyan Deng, and Zhao Kang. Multi-domain graph foundation models: Robust knowledge transfer via topology alignment. *arXiv preprint arXiv:2502.02017*, 2025.

[17] Haonan Yuan, Qingyun Sun, Junhua Shi, Xingcheng Fu, Bryan Hooi, Jianxin Li, and Philip S Yu. Graver: Generative graph vocabularies for robust graph foundation models fine-tuning. *arXiv preprint arXiv:2511.05592*, 2025.

[18] Junhua Shi, Qingyun Sun, Haonan Yuan, and Xingcheng Fu. Sa^{2}gfm: Enhancing robust graph foundation models with structure-aware semantic augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 15716–15724, 2026.

[19] Jiasheng Wu, Xin Liu, Dawei Cheng, Yi Ouyang, Xian Wu, and Yefeng Zheng. Safeguarding fraud detection from attacks: A robust graph learning approach. In *IJCAI*, pages 7500–7508, 2024.

[20] Michelle M Li, Kexin Huang, and Marinka Zitnik. Graph representation learning in biomedicine and healthcare. *Nature biomedical engineering*, 6(12):1353–1369, 2022.

[21] Yuhan Li, Yuyao Wang, Jianheng Tang, Heng Chang, Yuxiang Ren, and Jia Li. Advancing graph foundation models: A data-centric perspective. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 1635–1646, 2025.

[22] Zhongjian Zhang, Xiao Wang, Huichi Zhou, Yue Yu, Mengmei Zhang, Cheng Yang, and Chuan Shi. Can large language models improve the adversarial robustness of graph neural networks? In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pages 2008–2019, 2025.

[23] Jiayi Luo, Qingyun Sun, Haonan Yuan, Xingcheng Fu, and Jianxin Li. Robust graph learning against adversarial evasion attacks via prior-free diffusion-based structure purification. In *Proceedings of the ACM on Web Conference 2025*, pages 2098–2110, 2025.

[24] Woohyun Lee and Hogun Park. Self-supervised adversarial purification for graph neural networks. In *Forty-second International Conference on Machine Learning*, 2025.

[25] Yun Zhu, Haizhou Shi, Xiaotang Wang, Yongchao Liu, Yaoke Wang, Boci Peng, Chuntao Hong, and Siliang Tang. Graphclip: Enhancing transferability in graph foundation models for text-attributed graphs. In *Proceedings of the ACM on Web Conference 2025*, pages 2183–2197, 2025.

[26] Chunchun Chen, Xing Wei, Jiayi Yang, Chenrun Wang, Yiwei Fu, Yuxing Zhang, Xin Sun, Rui Fan, and Wei Ye. Edge self-adversarial augmentation enhances graph contrastive learning against neighborhood inconsistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 20005–20013, 2026.

[27] Chunchun Chen, Xing Wei, Yuxing Zhang, and Wei Ye. Adversarial augmentation with maximum discrepancy for graph contrastive learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

[28] Danny Wang, Ruihong Qiu, Guangdong Bai, and Zi Huang. Gold: Graph out-of-distribution detection via implicit adversarial latent generation. *arXiv preprint arXiv:2502.05780*, 2025.

[29] Daniel Zügner, Amir Akbarnejad, and Stephan Günnemann. Adversarial attacks on neural networks for graph data. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2847–2856, 2018.

[30] Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. *arXiv preprint arXiv:1902.08412*, 2019.

[31] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: An optimization perspective. *arXiv preprint arXiv:1906.04214*, 2019.

[32] Sixiao Zhang, Hongxu Chen, Xiangguo Sun, Yicong Li, and Guandong Xu. Unsupervised graph poisoning attack via contrastive loss back-propagation. In *Proceedings of the ACM Web Conference 2022*, pages 1322–1330, 2022.

[33] Simon Geisler, Tobias Schmidt, Hakan Şirin, Daniel Zügner, Aleksandar Bojchevski, and Stephan Günnemann. Robustness of graph neural networks at scale. *Advances in Neural Information Processing Systems*, 34:7637–7649, 2021.

[34] Susheel Suresh, Pan Li, Cong Hao, and Jennifer Neville. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34:15920–15933, 2021.

[35] Yeonjun In, Kanghoon Yoon, and Chanyoung Park. Similarity preserving adversarial graph contrastive learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 867–878, 2023.

[36] Kuan Li, YiWen Chen, Yang Liu, Jin Wang, Qing He, Minhao Cheng, and Xiang Ao. Boosting the adversarial robustness of graph neural networks: An ood perspective. In *The Twelfth International Conference on Learning Representations*, 2024.

[37] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 66–74, 2020.

[38] Xiang Zhang and Marinka Zitnik. Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in neural information processing systems*, 33:9263–9275, 2020.

[39] Kuan Li, Yang Liu, Xiang Ao, Jianfeng Chi, Jinghua Feng, Hao Yang, and Qing He. Reliable representations make a stronger defender: Unsupervised structure refinement for robust gnn. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 925–935, 2022.

[40] Yeonjun In, Kanghoon Yoon, Kibum Kim, Kijung Shin, and Chanyoung Park. Self-guided robust graph structure refinement. In *Proceedings of the ACM Web Conference 2024*, pages 697–708, 2024.

[41] Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 491–500, 2024.

[42] Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Language is all a graph needs. In *Findings of the association for computational linguistics: EACL 2024*, pages 1955–1973, 2024.

[43] Xi Zhu, Haochen Xue, Ziwei Zhao, Mingyu Jin, Wujiang Xu, Jingyuan Huang, Qifan Wang, Kaixiong Zhou, and Yongfeng Zhang. Llm as gnn: Graph vocabulary learning for graph foundation model. 2024.

[44] Zehong Wang, Zheyuan Zhang, Nitesh V Chawla, Chuxu Zhang, and Yanfang Ye. Gft: Graph foundation model with transferable tree vocabulary. *Advances in neural information processing systems*, 37:107403–107443, 2024.

[45] Yuhan Li, Peisong Wang, Zhixun Li, Jeffrey Xu Yu, and Jia Li. Zerog: Investigating cross-dataset zero-shot transferability in graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1725–1735, 2024.

[46] Gilbert W Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993.

[47] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning*, pages 9929–9939. PMLR, 2020.

[48] Shengyu Feng, Baoyu Jing, Yada Zhu, and Hanghang Tong. Ariel: Adversarial graph contrastive learning. *ACM Transactions on Knowledge Discovery from Data*, 18(4):1–22, 2024.

[49] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

[50] Xiaoxin He, Xavier Bresson, Thomas Laurent, Adam Perold, Yann LeCun, and Bryan Hooi. Harnessing explanations: Llm-to-lm interpreter for enhanced text-attributed graph representation learning. *arXiv preprint arXiv:2305.19523*, 2023.

[51] Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

[52] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 257–266, 2019.

[53] Xuanwen Huang, Kaiqiao Han, Yang Yang, Dezheng Bao, Quanjin Tao, Ziwei Chai, and Qi Zhu. Can gnn be good adapter for llms? In *Proceedings of the ACM web conference 2024*, pages 893–904, 2024.

[54] Jiarui Feng, Hao Liu, Lecheng Kong, Mingfang Zhu, Yixin Chen, and Muhan Zhang. Taglas: An atlas of text-attributed graph datasets in the era of large graph and language models. *arXiv preprint arXiv:2406.14683*, 2024.

[55] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

[56] Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, Weiwei Deng, Qi Zhang, Lichao Sun, Xing Xie, and Senzhang Wang. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. In *Advances in Neural Information Processing Systems*, 2023.

[57] Shujie Li, Yuxia Wu, Chuan Shi, and Yuan Fang. Hetgb: A comprehensive benchmark for heterophilic text-attributed graphs. *arXiv preprint arXiv:2503.04822*, 2025.

[58] Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew McCallum, Tom Mitchell, Kamal Nigam, and Sean Slattery. Learning to extract symbolic knowledge from the world wide web. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 509–516, 1998.

[59] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.

[60] Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Rémi Munos, Petar Veličković, and Michal Valko. Bootstrapped representation learning on graphs. In *ICLR 2021 workshop on geometrical and topological representation learning*, pages 1–14. OpenReview. net, 2021.

[61] Zhihao Wen and Yuan Fang. Augmenting low-resource text classification with graph-grounded pre-training and prompting. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 506–516, 2023.

[62] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. *arXiv preprint arXiv:2310.00149*, 2023.

[63] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.

A Theoretical Analysis

A.1 Notations

The key mathematical symbols and notations utilized in our problem formulation and the following proofs are listed in Table 4.

Table 4: Summary of Main Notations

Notation	Description
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$	Original text-attributed graph with nodes, edges, and texts
$N, \lvert\mathcal{E}\rvert$	Total number of nodes and edges in the graph
$\mathbf{A} \in \mathbb{R}^{N \times N}$	Original topological adjacency matrix
$\mathbf{X} \in \mathbb{R}^{N \times F}$	Node feature matrix encoded from raw text attributes
$G^i = (\mathbf{A}^i, \mathbf{X}^i)$	Localized ego-graph extracted around target node i
$\tilde{\mathbf{A}}$, $\tilde{\mathbf{D}}$	Symmetrically normalized adjacency matrix with self-loops and its degree matrix
$\mathcal{Y} = \{y_1, \ldots, y_C\}$	Set of candidate class labels for node classification
T_c	Semantic textual description for the c-th class
M	Number of fine-grained synthesized text anchors per class
T_c^m	The m-th specific synthesized text for class c
$f_\theta(\cdot)$, $g_\phi(\cdot)$	Pre-trained graph encoder (param θ) and text encoder (param ϕ)
$p_c(G)$	Predicted probability of graph G belonging to class c
$\mathbf{h}_i$	Latent hidden representation of node i after message passing
$\widehat{G}^i$	Purified ego-graph for the i-th target node.
$\widetilde{G}^i$	Adversarial ego-graph generated under topological attacks
$\mathcal{B}(G^i)$	Feasible search space of allowable perturbed topologies
Δ	Maximum allowable structural perturbation budget
$\mathbf{M}$	First-order Laplacian smoothed feature matrix ($\tilde{\mathbf{A}}\mathbf{X}$)
$H(G)$	Normalized smoothed spectral entropy for disorder measurement
τ	Diagnostic threshold triggering the defense recalibration
δ_{atk}	Adversarial structural perturbation
δ_{anti}	Learned discrete anti-attack structural perturbation
$\mathcal{L}$	Uniformity-guided optimization objective for purification
$\mathbf{S} \in \mathbb{R}^{N \times N}$	Continuous edge importance scoring matrix via gradient proxy
Δ_{add}, Δ_{drop}	Specific budgets for edge insertion and deletion operations

A.2 Proof of Proposition 1

Proposition 1 (Restated). *Maximizing the optimization objective $\mathcal{L}$ establishes the equivalence between simultaneously executing structural deviation from the adversarial input graph and performing structure-aware contrastive learning to preserve semantic consistency.*

Proof of Equivalence. Formally, the overall optimization objective $\mathcal{L}$ can be decoupled into two distinct functional components: an adversarial decoupling term $\mathcal{L}_{adv}$ and a semantic alignment term $\mathcal{L}_{sem}$.

First, $\mathcal{L}_{adv}$ corresponds to the first term of the objective, formulated as $-\cos(f_\theta(\widehat{G}), f_\theta(G))$. Maximizing $\mathcal{L}_{adv}$ is equivalent to minimizing the cosine similarity between the embedding of the purified graph $\widehat{G}$ and that of the distorted input G. This process explicitly forces the optimization trajectory to search for a structural topology that distinctly diverges from the adversarial input.

Second, $\mathcal{L}_{sem}$ comprises the second and the third terms of the objective, formulated as:

$$\mathcal{L}_{sem} = \cos\left(f_\theta(\widehat{G}), g_\phi(T_{ori})\right) - \log \sum_{c=1}^{C} \sum_{m=1}^{M} \exp\left(\cos\left(f_\theta(\widehat{G}), g_\phi(T_c^m)\right)\right). \tag{9}$$

In the standard contrastive learning paradigm, the InfoNCE loss aims to maximize the mutual information of a positive pair while pushing the query away from a set of negative samples:

$$\mathcal{L}_{InfoNCE} = -\log \frac{\exp(\cos(q, k_+)/\tau)}{\sum_{k \in K} \exp(\cos(q, k)/\tau)}. \tag{10}$$

Within the context of our framework, we formally define the purified graph embedding $q = f_\theta(\widehat{G})$ as the query, the original text attribute $k_+ = g_\phi(T_{ori})$ as the positive key, and the synthesized

fine-grained text descriptions $K = \{g_\phi(T_c^m)\}_{c=1,m=1}^{C,M}$ as the negative key pool. Substituting these exact definitions and a temperature parameter $\tau = 1$ into Eq. (10), and subsequently applying the logarithmic quotient rule to separate the numerator and the denominator, we obtain:

$$\mathcal{L}_{InfoNCE} = -\cos\left(f_\theta(\widehat{G}), g_\phi(T_{ori})\right) + \log \sum_{c=1}^{C} \sum_{m=1}^{M} \exp\left(\cos\left(f_\theta(\widehat{G}), g_\phi(T_c^m)\right)\right). \tag{11}$$

Comparing this result directly with Eq. (9) yields the strict equivalence $\mathcal{L}_{sem} \equiv -\mathcal{L}_{InfoNCE}$.

Consequently, maximizing the overall objective $\mathcal{L} = \mathcal{L}_{adv} + \mathcal{L}_{sem}$ operates as a dual-force optimization. It fundamentally guarantees that the purified graph breaks away from the distorted adversarial topology via $\mathcal{L}_{adv}$, while strictly preserving its original semantic consistency through the contrastive alignment enforced by $\mathcal{L}_{sem}$. This establishes the formal equivalence. □

A.3 Theoretical Justification and Implementation Details of Edge Scoring

In this section, we provide the formal mathematical proofs for the edge importance scoring strategy introduced in the main text, and we further elaborate on the fundamental engineering dilemma that necessitates our surrogate gradient approximation.

A.3.1 Formal Derivation of the Structural Gradient

To quantify the precise impact of topological perturbations on the optimization objective $\mathcal{L}$, we analyze the local message-passing dynamics.

Proposition 2 (First-Order Structural Gradient)**.** *Let $\mathcal{L}$ be the objective function to be maximized. Assume the aggregated representation of a root node i under a linear message-passing scheme is given by $\mathbf{h}_i = \sum_{k=1}^{N} \mathbf{A}_{ik}\mathbf{x}_k$, where $\mathbf{A}_{ik}$ denotes the adjacency state and $\mathbf{x}_k$ is the input feature of node k. The objective variation $\Delta\mathcal{L}$ induced by a discrete topological perturbation $\Delta\mathbf{A}_{ij} \in \{-1, 1\}$ is locally approximated by:*

$$\Delta\mathcal{L} \approx \left((\nabla_{\mathbf{h}_i}\mathcal{L})^\top \mathbf{x}_j\right) \Delta\mathbf{A}_{ij}$$

Proof. By applying the multivariate chain rule, the exact partial derivative of the objective $\mathcal{L}$ with respect to a scalar edge indicator $\mathbf{A}_{ij}$ is formulated as:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{A}_{ij}} = \left(\frac{\partial\mathcal{L}}{\partial\mathbf{h}_i}\right)^\top \frac{\partial\mathbf{h}_i}{\partial\mathbf{A}_{ij}}$$

Given the linear feature aggregation mechanism, computing the partial derivative of the hidden representation $\mathbf{h}_i$ with respect to $\mathbf{A}_{ij}$ isolates the specific neighbor's feature $\mathbf{x}_j$, because all other additive terms strictly vanish:

$$\frac{\partial\mathbf{h}_i}{\partial\mathbf{A}_{ij}} = \frac{\partial}{\partial\mathbf{A}_{ij}}\left(\sum_{k=1}^{N} \mathbf{A}_{ik}\mathbf{x}_k\right) = \mathbf{x}_j$$

Substituting this intermediate derivative back yields the exact structural gradient: $\frac{\partial\mathcal{L}}{\partial\mathbf{A}_{ij}} = (\nabla_{\mathbf{h}_i}\mathcal{L})^\top \mathbf{x}_j$. Finally, using a first-order Taylor expansion around the current adjacency state, the variation in the objective resulting from a single discrete edge flip $\Delta\mathbf{A}_{ij}$ is linearly bounded by:

$$\Delta\mathcal{L} \approx \frac{\partial\mathcal{L}}{\partial\mathbf{A}_{ij}}\Delta\mathbf{A}_{ij} = \left((\nabla_{\mathbf{h}_i}\mathcal{L})^\top \mathbf{x}_j\right) \Delta\mathbf{A}_{ij}$$

This concludes the proof. □

A.3.2 The Dilemma of Sparse Autograd and the Continuous Surrogate

While Proposition 2 provides the theoretically optimal trajectory for topological recalibration, it presents a critical computational paradox for modern graph neural networks.

The Disconnected Computational Graph. In state-of-the-art sparse GNN implementations (e.g., PyTorch Geometric[5] or DGL[6]), graph topology is strictly formatted as discrete integer indices (`edge_index`). The forward message-passing operates exclusively via *scatter/gather* operations along existing edges. Consequently, for any non-existent edge ($\mathbf{A}_{ij} = 0$), the nodes i and j never mathematically interact during the forward pass. The automatic differentiation (autograd) engine therefore cannot construct a computational pathway between them, rendering the true structural gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{A}_{ij}}$ strictly inaccessible for unobserved connections.

The $\mathcal{O}(N^2)$ Memory Wall. To forcefully extract gradients for non-edges, one would have to parameterize the entire graph as a dense continuous adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. This dense relaxation strategy immediately incurs an $\mathcal{O}(N^2)$ memory footprint, which inevitably triggers Out-Of-Memory (OOM) failures for standard real-world graphs and destroys the intrinsic efficiency of sparse text-attributed encoders.

The Surrogate Approximation. To resolve this bottleneck, our framework bypasses the uncomputable intermediate gradient $\nabla_{\mathbf{h}_i} \mathcal{L}$ and directly leverages the input feature gradient $\nabla_{\mathbf{x}_i} \mathcal{L}$. Because the input features $\mathbf{X} \in \mathbb{R}^{N \times F}$ are fully instantiated leaf nodes in the computational graph, $\nabla_{\mathbf{x}_i} \mathcal{L}$ is readily available with minimal $\mathcal{O}(N \times F)$ overhead. By utilizing the proxy score $\mathbf{S}_{ij} = (\nabla_{\mathbf{x}_i} \mathcal{L})^\top \mathbf{x}_j$, we successfully decouple the structural evaluation from dense topological parameterizations. This heuristic estimates the potential benefit of unobserved interactions, ensuring the anti-attack optimization remains scalable and memory-efficient strictly within the sparse domain.

B More Implementation Details

Adversarial Attack Settings. To comprehensively evaluate the robustness of our exact structural recalibration framework, we subject the zero-shot node classification task to state-of-the-art topological attacks. We strictly focus on the *test-time evasion setting*, where adversarial edges are injected or deleted during the inference phase without altering the base model's parameters. Specifically, we employ two highly representative attacks:

- *PGD (Projected Gradient Descent)* [31]: A classic and widely adopted first-order optimization attack. It utilizes gradient information from a surrogate model to iteratively generate continuous structural perturbations, which are subsequently projected back into the discrete topological space. PGD serves as a rigorous standard benchmark for evaluating worst-case structural vulnerabilities.
- *PRBCD (Projected Randomized Block Coordinate Descent)* [33]: A highly scalable and devastating attack specifically designed to overcome the memory bottlenecks of PGD on large-scale graphs. By leveraging randomized block coordinate descent, PRBCD efficiently explores the discrete combinatorial space to identify optimal edge flips, presenting an extreme challenge to test-time defense mechanisms.

To simulate varying degrees of adversarial threat, we evaluate the defense performance under progressive perturbation rates (e.g., the ratio of modified edges to the total original edges), ranging from low noise levels to severe structural destruction.

Ego-graph Sampling. Following the settings in [25], to ensure scalability when processing large-scale networks, we employ a Random Walk with Restart (RWR) strategy to sample the local structural context for each node. Specifically, rather than operating directly on the entire graph $\mathcal{G}$, we utilize the RWR sampler to extract a computationally tractable ego-graph G^i centered at each root node i. This process yields a collection of localized subgraphs $\{G^i\}_{i=1}^N$ that preserve the most relevant topological neighborhood while significantly reducing the computational overhead during inference.

Fine-grained Text Generation. To enrich the semantic representation of each class, we employ the `deepseek/deepseek-v4-pro`[7] model to generate fine-grained textual descriptions. We design dataset-specific prompts to guide the LLM in producing diverse, contextually relevant, and academically rigorous descriptions. For example, the prompt template utilized for the History dataset is illustrated below:

[5] https://pytorch-geometric.readthedocs.io/en/latest/
[6] https://www.dgl.ai/dgl_docs/
[7] https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro

Prompt for Fine-grained Text Generation (History Dataset)

Given the historical subject "<class_name>" with the basic description: "<base_description>", generate <num_descriptions> diverse and fine-grained descriptions that all start with "which". Each description should focus on specific historical events, periods, themes, or methodologies within the historical discipline.

Consider different aspects such as:

- Specific historical periods or eras
- Key historical figures and leaders
- Major events and turning points
- Cultural and social developments
- Economic and political transformations
- Wars and conflicts
- Technological and scientific advances
- Religious and philosophical movements
- Geographic and regional specializations
- Research methodologies and historical analysis approaches

Format: "which [specific historical topic, event, or theme in the <class_name> domain]"
Requirements:

- All <num_descriptions> descriptions must be unique and varied
- Each should explore a different aspect or subfield of the <class_name> historical category
- Focus on historical accuracy and scholarly relevance
- Ensure diversity in historical periods, geographic regions, and thematic approaches
- Make them specific to historical scholarship and educational contexts
- Emphasize academic rigor, methodological approaches, and evidence-based analysis

Generate exactly <num_descriptions> descriptions:

Based on the prompt $\mathcal{P}_{\text{generate}}$, the LLM successfully produces diverse and highly specific contexts. Table 5 showcases two representative generated samples for each category in the History dataset.

Table 5: Examples of fine-grained text descriptions generated by the LLM for the History dataset.

<class_name>	**Generated Fine-grained Descriptions**
Americas	• This book belongs to Americas, which explores the historical geography of the Caribbean's plantation islands and the environmental transformation for sugar monoculture. • This book belongs to Americas, which uses a global history approach to situate the American Revolution within a wider age of Atlantic revolutions.
Asia	• This book belongs to Asia, which traces the development of the Japanese writing system, including the adoption and adaptation of Chinese characters (kanji), and the invention of kana syllabaries. • This book belongs to Asia, which examines the international relations of the Tibetan Empire during the 7th-9th centuries, including its diplomatic and military engagements with Tang China and the Abbasid Caliphate.
Military	• This book belongs to Military, which investigates the political uses of history in Putin's Russia, including the commemoration of Victory Day and the reinterpretation of the Soviet past. • This book belongs to Military, which examines the history of Russian colonization in Alaska and the Pacific, and the sale of Alaska to the United States.
Africa	• This book belongs to Africa, which analyzes the rock art of the Tassili n'Ajjer and its insights into the Sahara's green period before desertification. • This book belongs to Africa, which explores the urban planning and stelae fields of the Aksumite Empire as markers of early Christian state formation in the Horn.

Evaluation Prompt Template. To transform the discrete class labels into natural language descriptions for zero-shot inference, we employ dataset-specific prompt templates. The purified feature embeddings are aligned with these semantic anchors to determine the final category. The detailed templates for each dataset are summarized in Table 6.

Table 6: Prompt templates used for transforming class labels into semantic descriptions.

Dataset	Prompt Template
Cora	this paper has a topic on $\{c\}$
CiteSeer	good paper of $\{c\}$
WikiCS	it belongs to $\{c\}$ research area
History	this book belongs to $\{c\}$
Children	the book is about $\{c\}$
Texas	this UT Austin computer science department page is a $\{c\}$ page
Cornell	this Cornell CS department page is a $\{c\}$ page
Washington	this university web page is a $\{c\}$ page
Wisconsin	this computer science department webpage is a $\{c\}$ page
Actor	this performer is classified as $\{c\}$

C Introduction of Datasets

Table 7: Statistics of datasets used to conduct experiments.

Dataset	Nodes	Edges	Class	Raw Text	Domain	Homo. Ratio
Cora	2,708	5,429	7	title and abstract of the paper	Academic	0.8100
CiteSeer	3,186	4,277	6	title and abstract of the paper	Academic	0.7437
WikiCS	11,701	215,863	10	webpage information	Wikipedia	0.6579
History	41,551	358,574	12	book description	E-commerce	0.6398
Children	76,875	1,162,522	24	book description	E-commerce	0.4043
Texas	187	328	5	webpage information	School Webpage	0.0643
Cornell	195	304	5	webpage information	School Webpage	0.1241
Washington	229	394	5	webpage information	School Webpage	0.1507
Wisconsin	265	510	5	webpage information	School Webpage	0.1808
Actor	4,416	12,172	5	actor biography	Wiki Webpage	0.5564

Along with the statistics in Table 7, we also give details of the datasets as follows:

- **Cora** [54] dataset comprises 2,708 scientific publications classified into seven machine learning categories: case-based, genetic algorithms, neural networks, probabilistic methods, reinforcement learning, rule learning, and theory. These categories serve as ground truth labels for downstream classification tasks. Furthermore, each paper in the citation network is connected to at least one other document, forming a total of 5,429 edges that capture the underlying structural dependencies essential for effective representation learning.
- **CiteSeer** [1] contains 3,186 scientific publications categorized into six distinct research domains: Agents, Machine Learning, Information Retrieval, Database, Human-Computer Interaction, and Artificial Intelligence. The primary objective of this benchmark is to accurately classify each paper into its respective domain by leveraging semantic features extracted from its title and abstract.
- **WikiCS** [55] serves as a benchmark dataset constructed from Wikipedia to evaluate graph neural networks. It encompasses ten distinct computer science disciplines exhibiting dense topological connections, with node representations derived directly from the textual content of the articles.
- **History** [56] is constructed from the Amazon Books collection. Nodes represent history-related books, and edges describe frequent co-purchase or co-view relationships between books. The classification objective is to predict the third-level book category, forming a 12-class node classification task. Each node is equipped with textual attributes derived from the book title and description, enabling the model to exploit both semantic content and graph connectivity.
- **Children** [56] originates from the Amazon-Books dataset, restricted to items carrying the second-level label Children. Each node denotes a specific book, utilizing its title and description as textual features. Edges capture frequent co-purchase or co-viewing interactions between books. The classification task requires assigning these nodes to 24 categories defined by their third-level labels.

- **Texas** [57] is a WebKB-based text-attributed webpage graph collected from the University of Texas computer science department. Each node represents a webpage, and each edge denotes a hyperlink between two webpages. The task is to classify webpages into five categories, including student, project, course, staff, and faculty. The textual attribute of each node is extracted from the corresponding webpage content.
- **Cornell** [57] is a WebKB-based text-attributed webpage graph collected from the Cornell University computer science department. Nodes correspond to webpages, and edges represent hyperlinks among these webpages. The node classification task follows the same five-category webpage classification setting, including student, project, course, staff, and faculty. Each node is associated with raw webpage text.
- **Washington** [58] is a WebKB-style school webpage graph collected from the University of Washington computer science department. Each node denotes a webpage, and edges capture hyperlink relations between webpages. The classification task aims to assign each webpage to a semantic webpage type, such as student, faculty, staff, course, and project. In our experiments, each node is associated with webpage textual information, making it suitable for zero-shot text-attributed graph evaluation.
- **Wisconsin** [57] is a WebKB-based text-attributed webpage graph collected from the University of Wisconsin computer science department. Similar to the other WebKB-style datasets, nodes represent webpages and edges represent hyperlinks. The task is to classify each webpage into five categories: student, project, course, staff, and faculty. The graph is naturally heterophilic because linked webpages do not necessarily belong to the same class.
- **Actor** [57] is an actor-centered text-attributed social graph derived from a film-director-actor-writer network. Each node represents an actor, and an edge indicates that two actors co-occur on the same Wikipedia page. The node text is based on actor-related biographical and categorical information, with category-related terms removed to avoid label leakage. The classification task considers five actor groups, including American film actors, American film and television actors, American television and stage actors, English actors, and Canadian actors.

D Introduction of Baselines

We provide a brief introduction to the compared baseline methods as follows:

- **DGI** [59] learns node representations in an unsupervised manner by maximizing the mutual information between local patch representations and high-level graph summaries. By avoiding traditional random walk objectives, it provides a general approach applicable to both transductive and inductive learning setups. Code: https://github.com/PetarV-/DGI
- **BGRL** [60] introduces a self-supervised approach that eliminates the need for negative pairs, avoiding the quadratic computational bottleneck typical of contrastive learning. Inspired by BYOL, it enables the effective use of graph attentional encoders and matches or outperforms previous unsupervised state-of-the-art methods. Code: https://github.com/nerdslab/bgrl
- **G2P2** [61] addresses low-resource text classification through a two-pronged approach of graph-grounded pre-training and downstream prompting. By jointly pre-training a graph-text model using contrastive strategies, it effectively leverages inherent network structures to achieve strong zero- and few-shot performance. Code: https://github.com/WenZhihao666/G2P2
- **GraphGPT** [41] integrates large language models with graph structural knowledge using a text-graph grounding component and a lightweight alignment projector. Through a dual-stage instruction tuning process, it achieves exceptional zero-shot generalization across diverse graph learning tasks without relying on downstream labels. Code: https://github.com/HKUDS/GraphGPT
- **OFA** [62] unifies diverse graph datasets and tasks into a single framework by representing nodes and edges with natural language descriptions encoded by language models. It introduces a novel graph prompting paradigm that appends prompting substructures to the input graph, enabling general-purpose, cross-domain classification in zero-shot and few-shot scenarios without fine-tuning. Code: https://github.com/LechengKong/OneForAll
- **ZeroG** [45] addresses cross-dataset zero-shot transferability by utilizing a language model to encode node attributes and class semantics into a consistent feature space. It employs a prompt-based

subgraph sampling module and a lightweight fine-tuning strategy to enrich structural information while mitigating negative transfer and overfitting. Code: https://github.com/NineAbyss/ZeroG

- **GraphCLIP** [25] addresses the limitations of cross-domain transferability in text-attributed graphs through a self-supervised contrastive graph-summary pretraining method. By leveraging LLM-generated graph-summary pairs and introducing a novel graph prompt tuning technique, it achieves strong zero-shot and few-shot performance while mitigating catastrophic forgetting. Code: https://github.com/zhuyun97/graphclip

E Algorithms

The complete execution flow of our proposed framework is summarized in Algorithm 1.

Algorithm 1: The test-time structural recalibration paradigm of TRACE.

Input: Test ego-graph $G = (\mathbf{A}, \mathbf{X})$, pre-trained graph encoder f_θ, text encoder g_ϕ, text labels $\{T_c\}_{c=1}^{C}$, synthesized texts $\{T_c^m\}_{m=1}^{M}$, entropy threshold τ, edge modification budgets Δ_{add} and Δ_{drop}, number of steps E.

// Phase 1: Selective Recalibration Trigger

Compute normalized smoothed feature matrix $\mathbf{M} = \tilde{\mathbf{A}}\mathbf{X}$
Perform SVD on $\mathbf{M}$ to compute normalized spectral entropy $H(G)$

if $H(G) \leq \tau$ **then**
  return Original graph G and prediction based on Eq. (1)
else if $H(G) > \tau$ **then**
  // Phase 2: Initialization with Raw Test Graph
  Initialize purified graph $\widehat{G}^0 \leftarrow G$ with adjacency $\widehat{\mathbf{A}}^0 = \mathbf{A}$
  Compute constant semantic anchors using frozen g_ϕ
  Detach initial graph embedding $f_\theta(G)$ as a adversarial anchor
  Enable gradient tracking for continuous feature matrix $\mathbf{X}$
  // Phase 3: E step Iterative Purification
  for $e = 1, 2, \ldots, E$ **do**
    Compute objective $\mathcal{L}^e$ on current graph $\widehat{G}^{e-1}$ using Eq. (5)
    Compute input feature gradients $\nabla_{\mathbf{X}}\mathcal{L}^e$ via backpropagation
    Approximate structural scoring matrix $\mathbf{S}^e = (\nabla_{\mathbf{X}}\mathcal{L}^e)\mathbf{X}^\top$
    // Edge Insertion and Deletion updates
    Select non-edges with top-(Δ_{add}/N) highest positive scores in $\mathbf{S}^e$
    Select existing edges with top-(Δ_{drop}/N) most negative scores in $\mathbf{S}^e$
    Update $\widehat{\mathbf{A}}^e$ by applying the selected edge flips
    Update ego-graph $\widehat{G}^e \leftarrow (\widehat{\mathbf{A}}^e, \mathbf{X})$
  // Final Inference
  return Purified graph $\widehat{G}^E$ and zero-shot prediction $p_c(\widehat{G}^E)$

F More Experiment Results

Complexity Analysis. The complexity of TRACE during test-time inference is primarily governed by three components. Let N and $|\mathcal{E}|$ be the number of nodes and edges in a localized ego-graph, respectively. F denotes the input feature dimension, D the hidden dimension, and E the number of optimization iterations.

Selective Triggering: Computing the normalized smoothed spectral entropy requires $O(|\mathcal{E}|F)$ for Laplacian smoothing and $O(NFk)$ for truncated SVD (where k is the number of singular values). This step is non-parametric and executed only once.

Recalibration Objective: Given that LLM-based semantic anchors are pre-computed, calculating the uniformity-guided loss $\mathcal{L}$ costs $O(MCD)$ per iteration, where C is the number of classes and M is the number of descriptions per class.

Structural Optimization: The first-order gradient approximation via sparse GNNs takes $O(E(|\mathcal{E}|D + ND^2))$. The edge importance scoring matrix $\mathbf{S}$ requires $O(EN^2F)$, followed by $O(EN^2 \log N^2)$ for edge flipping.

The total complexity is $O(|\mathcal{E}|F+NFk+E(MCD+|\mathcal{E}|D+ND^2+N^2F))$. Since TRACE operates on localized ego-graphs ($N \ll N_{global}$) and only triggers optimization for adversarial candidates, it maintains high efficiency compared to global graph defense baselines.

More Spectral Entropy H Quantitative Analysis. Figure 6 illustrates the distribution of spectral entropy (H) on the History and WikiCS datasets under varying intensities of PGD and PRBCD attacks. Spectral entropy monotonically increases across all scenarios as the perturbation budget scales from 0% (clean) to 40%. For example, under a PRBCD attack on the History dataset, average spectral entropy rises from 0.690 on clean graphs to 0.800 at a 40% perturbation rate. The violin plots capture a distinct distributional shift: increasing topological noise concentrates the probability density of H at higher values, reflecting the progressive destruction of intrinsic structural regularities. Under identical perturbation budgets, PRBCD attacks induce a more pronounced entropy increase than PGD attacks. These observations confirm spectral entropy as a sensitive and reliable non-parametric metric for quantifying adversarial topological distortions.

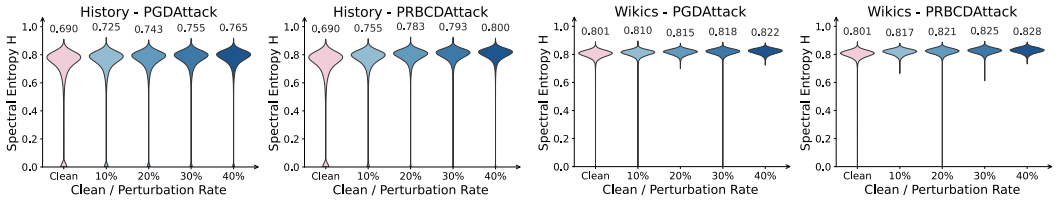

Figure 6: More quantitative analysis of spectral entropy H on History and WikiCS datasets.

G Broader Impacts and Limitations

This study addresses the challenge of robust zero-shot inference at test-time under adversarial attacks, which has significant potential for advancing the secure deployment of graph foundation models in security-critical applications by reducing the computational overhead associated with adversarial retraining. However, it is important to acknowledge that this work represents an initial exploration in test-time structural recalibration and may have certain limitations. For instance, our method may not perform optimally in complex real-world scenarios involving joint perturbations, such as simultaneous attacks on both graph topology and input text, which could compromise the generated semantic anchors and mislead the recalibration process. Additionally, this study relies on the semantic priors provided by a frozen pre-trained text encoder. In specialized domains where the pre-trained model lacks sufficient knowledge to generate discriminative text anchors, the performance gains of our uniformity-guided optimization may be constrained. Furthermore, our current empirical evaluation is primarily anchored on GraphCLIP due to the limited public availability of foundation model checkpoints. Although our framework is model-agnostic, its generalizability across diverse architectural backbones requires further empirical validation. In future research, our goal is to address these challenges by evaluating our method on a wider array of open-source models and expanding our test-time recalibration paradigm to encompass more generalized adversarial defense scenarios.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have made clear claims about contributions and scopes in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations part is discussed.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide detailed theoretical proofs in the main paper and appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We describe the details of our model and training strategy in the paper for reproduction.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide a link to the code in the implementation details.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have introduced experiment settings and details, and added more details in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We precisely defined and reported the error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All the experiments are conducted on a single NVIDIA 3090 24GB GPU.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We sincerely read the ethics guidelines and obey this rule.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The discussion about the broader impacts is provided.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate

Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We explicitly cited the sources of the relevant data and other materials used in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: New assets are well documented.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: In this paper, our model leverages large language models (LLMs) to generate fine-grained textual descriptions.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.