BEAST: BOUNDARY-AWARE SEMANTIC EXTRAPOLATION WITH DEBIASED TOPOLOGY FOR LONG-TAILED TEXT-ATTRIBUTED GRAPH LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper studies the problem of long-tailed text-attributed graph learning, where severe class imbalance hinders the learning of tail classes due to limited semantic evidence and biased structural contexts. Existing approaches typically augment minority classes to create a balanced graph and train graph neural networks (GNNs) on the augmented data. Despite the progress, their performance is far from satisfactory due to overconcentration among synthesized tail nodes and noisy edges during graph augmentation. Towards this end, we propose a novel framework named Boundary-aware Semantic Extrapolation with Debiased Topology (**BEAST**) for long-tailed text-attributed graph learning. The core of our **BEAST** is to generate informative nodes along with reliable edges in augmented graphs from two complementary perspectives, i.e., *semantic boundary exploration* and *structural pattern acquisition*. In particular, we first identify semantic class competitors of each tail class by measuring inter-class prototype distance in the latent space. More importantly, we leverage large language models (LLMs) to generate auxiliary subclasses for tail classes, guiding node synthesis toward desired regions while maintaining sufficient separation from competing classes. These auxiliary subclasses would ensure diversity during extrapolation, while boundary-aware filtering guarantees semantic consistency of synthesized nodes. In addition, we sample a class-balanced graph as a calibration set to train a shadow model for predicting connectivity patterns, which are subsequently utilized for debiased topology imputation. To mitigate parent-child competition, we further adjust the prediction logits and optimize the model in a multi-task learning framework. Extensive experiments on multiple long-tailed text-attributed graph benchmarks demonstrate the effectiveness of **BEAST** against representative baselines. Our code is available at https://anonymous.4open.science/r/BEAST-CC22.

1 INTRODUCTION

Graph Neural Networks (GNNs) have become a dominant approach for modeling and learning from graph-structured data, with broad application in citation networks (Lin et al., 2024; Kipf & Welling, 2016), recommendation systems (Ying et al., 2018; Guo et al., 2022) and social networks (Zhang et al., 2022; Guo et al., 2022). Among various graph learning tasks, node classification is one of the most fundamental, aiming to predict node labels by exploiting both node attributes and structural information. In many real-world graphs, node attributes are naturally expressed in the form of rich textual descriptions, bringing additional semantic information to graph representation learning that complements structural information (Hormazabal et al., 2022).

The integration of textual semantics with graph structure gives rise to text-attributed graphs (TAGs), where each node is associated with textual attributes in addition to graph topology (Zhao et al., 2022; Yan et al., 2023). In real-world TAGs, class distributions are often highly long-tailed, with a few head classes containing abundant labeled nodes while tail classes are severely underrepresented (Park et al., 2021). This imbalance makes it difficult to learn representations for minority classes, as both structural patterns and textual semantics can be dominated by head classes. For example, recommendation graphs combine product text with user-item interactions, where popular categories often dominate niche ones in both frequency and connectivity (Wu et al., 2019).

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

To address long-tailed TAG learning, existing studies mainly explore data rebalancing and augmentation strategies to address insufficient supervision for minority classes (Zhang et al., 2024). Conventional strategies include re-sampling to balance class distributions (Hu et al., 2025; Zhou & Gong, 2023), re-weighting to emphasize tail classes (Song et al., 2022; Chen et al., 2021), and interpolation-based augmentation to synthesize minority representations (Wu et al., 2022; Qin et al., 2025). Nonetheless, these methods primarily operate at the data or representation level and do not directly enrich the textual semantics of tail classes (Ma et al., 2025). Recently, large language models (LLMs) have provided new opportunities for semantic-level augmentation by generating textual information for minority classes (Wang et al., 2025a; Zhao et al., 2024). Such approaches demonstrate the potential of leveraging LLMs for long-tailed TAG learning.

While LLMs offer a promising way to enrich the semantic information of minority classes, long-tailed TAG learning requires both semantic discrimination and structural dependencies (Jin et al., 2024; Li et al., 2025). Therefore, how to effectively integrate LLM-based augmentation into long-tailed TAG learning remains nontrivial, giving rise to two key challenges: ❶ ***Limited discriminative boundary of tail classes.*** Tail classes exhibit insufficient semantic diversity, leaving their underlying variations and ambiguous regions poorly characterized and making discriminative representations difficult to learn. ❷ ***Biased structural context under long-tailed distribution.*** Class imbalance favors dominant classes during graph learning, where skewed neighborhoods further amplify head-class signals and weaken informative evidence from tail classes. As shown in Figure 1, these challenges reveal that long-tailed TAG learning suffers from both limited semantic discrimination and biased structural information, motivating a unified solution from both perspectives.

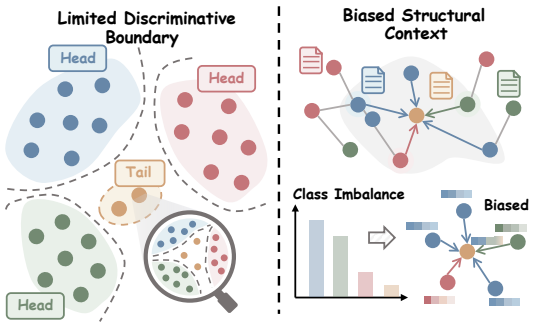


Figure 1: Motivation of our BEAST. Long-tailed TAGs suffer from limited tail-class discrimination and biased structural context, where head-class signals dominate neighborhood aggregation and bias tail-node representations.

To address the aforementioned challenges, we propose a Boundary-aware Semantic Extrapolation with Debiased Topology for Long-Tailed Text-attributed Graph Learning (BEAST). The key idea of BEAST is to enrich tail classes with diverse boundary semantics while reliably integrating the extrapolated information into graph learning. Toward this end, we propose fine-grained auxiliary subclasses by identifying semantic competitors of each tail class, which provide informative directions for semantic exploration. Guided by these subclasses, we extrapolate category-aware nodes toward ambiguous target–competitor boundaries and retain subclass-consistent samples. We then learn unbiased connectivity patterns from a class-rebalanced shadow graph to impute reliable topology for the generated nodes. Finally, we formulate hierarchical supervision as a multi-task learning framework that performs subclass-level discrimination and parent-level classification, preserving fine-grained auxiliary semantics while maintaining consistency with the original label space.

This paper's contributions are summarized as follows: ❶ ***New Perspective:*** We revisit long-tailed TAG learning from both semantic granularity and topology completion, highlighting the importance of capturing fine-grained tail-class semantics and recovering informative structural dependencies. ❷ ***New Methodology:*** We propose BEAST, which enriches tail classes beyond the observed data and integrates the resulting semantic evidence into graph structure, with hierarchical supervision further facilitating effective long-tailed TAG learning. ❸ ***Extensive Experiments:*** We conduct extensive experiments on multiple long-tailed TAG benchmark datasets, where BEAST consistently outperforms representative baselines and and maintains strong performance across different evaluation metrics.

2 Related Work

Long-Tailed Learning on Graphs. Long-tailed distributions widely exist in real-world graphs, causing GNNs to favor head classes and underrepresent tail classes (Liu et al., 2025). Existing solutions mainly rely on data augmentation or specialized learning strategies (Wang et al., 2026a). Data-level methods synthesize minority nodes, representations, or local structures, including GraphSMOTE (Zhao et al., 2021), GraphMixup (Wu et al., 2022), GraphENS (Park et al.,

2021), and GraphSHA (Li et al., 2023). Model-level approaches instead improve long-tailed learning through tailored architectures or objectives, such as the expert-based LTE4G (Yun et al., 2022) and hierarchical learning in HierTail (Wang et al., 2024). Despite their effectiveness, these methods operate around observed minority distributions, leaving ambiguous semantic regions underexplored. **BEAST** addresses this limitation through semantic extrapolation coupled with topology imputation.

LLM-Based Augmentation for Text-Attributed Graphs. The rich textual attributes of TAGs have encouraged the integration of LLMs into graph representation learning (Jin et al., 2024; Su et al., 2025). Existing studies mainly use LLMs for semantic encoding, augmentation, or knowledge enhancement (Zhang et al., 2024). GAugLLM (Fang et al., 2024), LATEX-GCL (Yang et al., 2024), CTGL (Zhou et al., 2025), and TANS (Wang et al., 2025b) enhance textual or structural representations through LLM-assisted augmentation, while UniS^{2}A (Pan et al., 2026) jointly models semantic and structural augmentation and SAVE-TAG (Wang et al., 2026b) targets long-tailed TAGs via LLM-based interpolation. However, auxiliary semantic exploration and structural integration of nodes remain limited, motivating **BEAST** to combine semantic extrapolation with topology imputation.

3 The Proposed BEAST

Preliminary. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ be a text-attributed graph, where $\mathcal{V}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denote the node and edge sets, respectively, and $\mathcal{T} = \{t_i\}_{i \in \mathcal{V}}$ denotes the associated node texts. We denote the graph adjacency matrix by $\boldsymbol{A} \in \{0,1\}^{|\mathcal{V}| \times |\mathcal{V}|}$ and its normalized form by $\widehat{\boldsymbol{A}}$. Each node $v_i \in \mathcal{V}$ is associated with a label $y_i \in \mathcal{Y} = \{1, \ldots, C\}$. A frozen text encoder $\phi(\cdot)$ maps each node text t_i into a semantic representation $\boldsymbol{h}_i = \phi(t_i)$, with $\boldsymbol{h}_i \in \mathbb{R}^d$. For each class $c \in \mathcal{Y}$, let n_c denote the number of labeled nodes belonging to class c, and we use c_t to denote a tail class. The imbalance ratio is defined as $\mathrm{IR} = n_{\min}/n_{\max}$, where $n_{\max} = \max_{c \in \mathcal{Y}} n_c$ and $n_{\min} = \min_{c \in \mathcal{Y}} n_c$. Accordingly, a smaller IR corresponds to a more imbalanced class distribution. Given the long-tailed graph $\mathcal{G}$, the node prediction model is formulated as $F_{\mathrm{cls}}(\cdot) = f_{\varpi}(g_{\vartheta}(\cdot))$, where g_{ϑ} denotes the GNN encoder and f_{ϖ} denotes the node classifier. Specifically, g_{ϑ} maps graph-structured inputs to node representations, while f_{ϖ} produces predictions over the original label space $\mathcal{Y}$. Our goal is to learn F_{cls} for accurate node classification under the long-tailed setting.

3.1 Framework Overview

In this work, we focus on long-tailed learning on text-attributed graphs. To alleviate the insufficient semantic and structural evidence of tail classes, we propose **BEAST**, a unified framework which enrich tail-class evidence through semantic expansion and topology imputation, while exploiting the induced hierarchical semantics for effective learning. As illustrated in Figure 2, given a long-tailed TAG, we first identify the neighbor categories of each tail class and leverage LLMs to discover auxiliary subclasses, which are then used to guide boundary-oriented node generation. To integrate these generated nodes into graph propagation, we construct a class-rebalanced shadow graph and learn structural recovery signals by training a debiased shadow model, thereby recovering plausible connections for the augmented nodes. Finally, we train the GNN on the resulting augmented graph with hierarchical supervision, preserving the fine-grained semantics of auxiliary categories while maintaining their consistency with the original parent classes.

3.2 Auxiliary Subclass Proposal for Diverse Semantics

Tail classes often exhibit limited semantic diversity due to sparse observations, making their underlying variations insufficiently characterized (Zhao et al., 2025). To expose more diverse semantic directions, we are motivated to seek auxiliary subclasses that capture fine-grained variations of each tail class while remaining distinguishable from its semantic competitors. We first encode each node text t_i with a frozen text encoder $\phi(\cdot)$ to obtain its representation $\boldsymbol{h}_i = \phi(t_i)$. Based on these embeddings, we compute a prototype for each class c as a coarse semantic anchor for subsequent competitor retrieval and boundary screening:

$$\boldsymbol{\mu}_c = \frac{\frac{1}{n_c}\sum_{i \in \mathcal{I}_c} \boldsymbol{h}_i / \|\boldsymbol{h}_i\|_2}{\left\| \frac{1}{n_c}\sum_{i \in \mathcal{I}_c} \boldsymbol{h}_i / \|\boldsymbol{h}_i\|_2 \right\|_2}, \ \mathcal{I}_c = \{i \mid y_i = c\}, \tag{1}$$

where $\mathcal{I}_c$ denotes the index set of samples belonging to class c, and $n_c = |\mathcal{I}_c|$. To capture informative semantic directions around each tail class c_t, we retrieve its q most semantically similar competitors

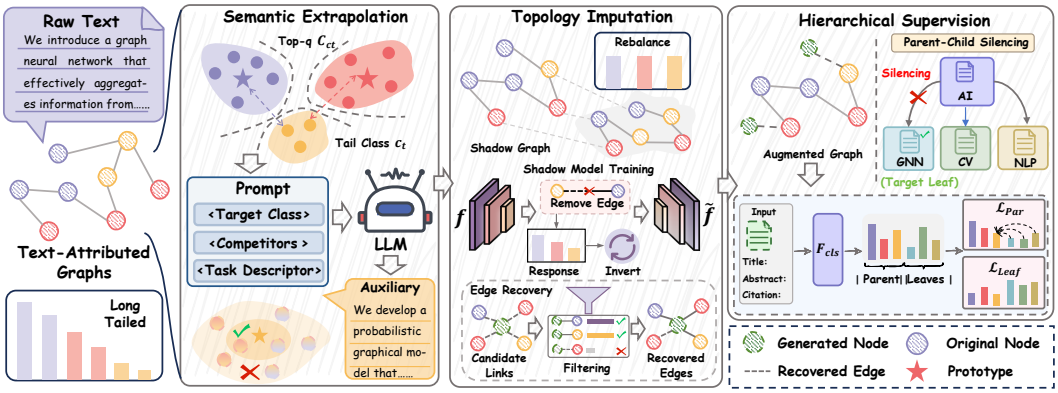


Figure 2: Framework overview of our **BEAST**. We introduce auxiliary subclasses to guide boundary-aware semantic extrapolation, restore reliable topology for generated nodes through connectivity patterns learned from a rebalanced shadow graph, and leverage hierarchical multi-task supervision over subclass and parent labels for effective long-tailed TAG learning.

according to prototype similarity:

$$\mathcal{C}_{c_t} = \underset{c \neq c_t}{\arg\operatorname{topq}} \cos(\boldsymbol{\mu}_{c_t}, \boldsymbol{\mu}_c), \tag{2}$$

where $\mathcal{C}_{c_t}$ contains the q semantically closest non-target classes to c_t. Conditioned on the target class name, its competitor class names, and a short task descriptor D_{task}, we utilize the LLM to propose fine-grained auxiliary subclasses for each tail class c_t. Each proposal is represented as:

$$\omega_{c_t m} = (k_{c_t m}, s_{c_t m}) \sim p_{\text{LLM}}\left(\cdot \mid D_{\text{task}}, \text{name}(c_t), \{\text{name}(c) : c \in \mathcal{C}_{c_t}\}\right), \tag{3}$$

where $k_{c_t m}$ and $s_{c_t m}$ denote the generated keyword and description, respectively. We retain the proposals that provide the controlled semantic variations yet separated from the target class:

$$\ell_{\text{cat}} \leq d_{c_t}(\boldsymbol{h}_{c_t m}) \leq u_{\text{cat}} \qquad d_{c_t}(\boldsymbol{h}_{c_t m}) < d_{\mathcal{C}}(\boldsymbol{h}_{c_t m}), \tag{4}$$

where $\boldsymbol{h}_{c_t m} = \phi(k_{c_t m} \| s_{c_t m})$ is the proposal embedding, d_{c_t} and $d_{\mathcal{C}}$ denote the cosine distances to the target and nearest competitor prototypes, and ℓ_{cat} and u_{cat} specify the admissible semantic expansion range. The accepted proposals form the auxiliary subclass set $\mathcal{A}_{c_t}$ for tail class c_t.

3.3 Semantics Extrapolation with Boundary-aware Filtering

To enrich underrepresented tail regions, we extrapolate semantics with auxiliary subclasses and retain boundary-aware samples through filtering. Specifically, we generate subclass-conditioned nodes close to the target–competitor boundary while remaining on the target side.

For each tail class c_t, we define its class deficit as $B_{c_t} = n_{\max} - n_{c_t}$, where $n_{\max} = \max_c n_c$, and generate $N_{c_t}^{\text{cand}} = \lceil \gamma B_{c_t} \rceil$ candidate nodes with $\gamma \geq 1$. The generation budget is approximately evenly distributed across the auxiliary subclasses in $\mathcal{A}_{c_t}$. For each auxiliary subclass $\omega_{c_t m} = (k_{c_t m}, s_{c_t m})$, candidate texts are generated as:

$$\widetilde{t}_{c_t j} \sim p_{\text{LLM}}\left(\cdot \mid k_{c_t m}, s_{c_t m}, \text{name}(c_t), \{\text{name}(c) : c \in \mathcal{C}_{c_t}\}, \mathcal{T}_{c_t}^{\text{ref}}, D_{\text{task}}\right), \tag{5}$$

where $\mathcal{T}_{c_t}^{\text{ref}}$ contains representative texts from the target class, and γ controls the degree of semantic extrapolation. To identify generated nodes that enrich ambiguous decision regions, we measure their proximity to the target–competitor boundary. Specifically, we define the boundary margin as:

$$\Delta_{c_t}(\boldsymbol{h}) = d_{\mathcal{C}}(\boldsymbol{h}) - d_{c_t}(\boldsymbol{h}), \tag{6}$$

where d_{c_t} and $d_{\mathcal{C}}$ are the cosine distances to the target and nearest competitor prototypes, respectively. A small positive margin indicates proximity to the target–competitor boundary while remaining on the target side. Accordingly, for each generated text $\widetilde{t}_{c_t j}$ with representation $\widetilde{\boldsymbol{h}}_{c_t j} = \phi(\widetilde{t}_{c_t j})$, we retain only candidates satisfying:

$$0 < \Delta_{c_t}(\widetilde{\boldsymbol{h}}_{c_t j}) \leq \tau_{\text{bd}}, \tag{7}$$

where $\tau_{\text{bd}} > 0$ controls the admissible proximity to the semantic boundary. We further verify them by an LLM critic $C_{\text{node}}(\widetilde{t}_{c_t j}, \omega_{c_t m}) \in \{0, 1\}$ for auxiliary subclass consistency. The accepted nodes inherit their auxiliary subclass labels and form $\mathcal{V}^+$.

3.4 Unbiased Topology Imputation from Connectivity Patterns

Although semantic augmentation enriches tail-class nodes, the generated nodes require structural connections for effective message passing. Inspired by confidence-response modeling for graph connectivity inference (Zhang et al., 2026), we learn structural evidence from controlled edge removal on a class-rebalanced shadow graph to recover plausible connections for $\mathcal{V}^+$.

Debiased Shadow Learning for Structural Evidence Discovery. To reduce the structural bias inherited from the original long-tailed graph, we first construct a balanced shadow graph $\mathcal{G}^{\mathrm{sh}} = (\mathcal{V}^{\mathrm{sh}}, \mathcal{E}^{\mathrm{sh}})$ via class-aware adaptive pooling. Specifically, nodes are sampled according to:

$$\pi_i \propto \frac{1}{n_{y_i}^{\alpha}} \exp\left(\beta\, \boldsymbol{\xi}^{\top} \boldsymbol{h}_i\right), \qquad \alpha \in [0,1], \tag{8}$$

where α controls class rebalancing, β scales semantic preference, and $\boldsymbol{\xi}$ promotes intra-class diversity. The induced edges among sampled nodes form $\mathcal{E}^{\mathrm{sh}}$. We first train a GNN f_θ on $\mathcal{G}^{\mathrm{sh}}$ and remove a sampled edge subset $\mathcal{D}_u \subset \mathcal{E}^{\mathrm{sh}}$ to expose prediction responses associated with structural perturbations. The resulting perturbed model $\widetilde{f}$ is obtained as:

$$\widetilde{\theta} = \theta - \eta \nabla_\theta \mathcal{L}_{\mathrm{CE}}\left(\mathcal{G}^{\mathrm{sh}} \setminus \mathcal{D}_u\right), \qquad \widetilde{f} = f_{\widetilde{\theta}}. \tag{9}$$

where η is the update step size, $\mathcal{L}_{\mathrm{CE}}$ is the cross-entropy loss, and $\widetilde{\theta}$ denotes the parameters after edge removal. To further capture how local confidence responds to structural perturbation, we encode its two-hop propagation response as:

$$\delta_i^{(\ell)} = \zeta_i^{(\ell)} - \zeta_i^{(\ell-1)}, \qquad \boldsymbol{\chi}_i = \bigoplus_{\ell=1}^{2} \left[\delta_i^{(\ell)} \oplus \frac{\delta_i^{(\ell)}}{\zeta_i^{(\ell-1)} + \epsilon}\right], \quad \boldsymbol{\zeta}^{(\ell)} = \widehat{\boldsymbol{A}}_{-} \boldsymbol{\zeta}^{(\ell-1)}. \tag{10}$$

where $\zeta_i^{(0)} = \max_k \rho_{i,k}$ is the initial confidence, $\widehat{\boldsymbol{A}}_{-}$ is the normalized adjacency after removing $\mathcal{D}_u$, and $\boldsymbol{\chi}_i$ aggregates the absolute and relative confidence responses $\delta_i^{(\ell)}$, with ϵ for numerical stability. Based on the post-removal predictions and confidence trends, we characterize the confidence similarity gap between node pairs together with their semantic and local structural responses:

$$\boldsymbol{a}_{ij} = |\boldsymbol{\rho}_i - \boldsymbol{\rho}_j| \oplus (\boldsymbol{\rho}_i \odot \boldsymbol{\rho}_j) \oplus \cos(\boldsymbol{h}_i, \boldsymbol{h}_j) \oplus (\boldsymbol{\chi}_i \odot \boldsymbol{\chi}_j), \tag{11}$$

where $\boldsymbol{\rho}_i = \widetilde{f}(\mathcal{G}^{\mathrm{sh}} \setminus \mathcal{D}_u)_i$ denotes the post-removal posterior. We then learn a edge recovery predictor $g_\psi(\boldsymbol{a}_{ij}) = \sigma(\mathrm{MLP}_\psi(\boldsymbol{a}_{ij}))$, using retained and unlearned edges as positive pairs while sampled non-edges as negatives. The corresponding training objective is:

$$\mathcal{L}_{\mathrm{inv}}^{\boldsymbol{\psi}} = -\frac{1}{|\mathcal{Q}|} \sum_{(i,j)\in\mathcal{Q}} \left[m_{ij} \log g_\psi(\boldsymbol{a}_{ij}) + (1 - m_{ij}) \log(1 - g_\psi(\boldsymbol{a}_{ij}))\right], \tag{12}$$

where $m_{ij} = \mathbf{1}\{(i,j) \in \mathcal{E}^{\mathrm{sh}}\}$ denotes the edge-membership label, and $\mathcal{Q}$ contains sampled positive and negative node pairs. Together, $\widetilde{f}$ and g_ψ form the debiased shadow model $F_{\mathrm{sh}}(\cdot) = g_\psi(\widetilde{f}_{\widetilde{\theta}}(\cdot))$.

Structural Response-guided Graph Structure Refinement. Given the trained shadow model F_{sh}, we leverage structural evidence to recover reliable connections for the augmented nodes. To construct a provisional topology, we obtain $\boldsymbol{h}_v = \phi(\widetilde{t}_{c_t j})$ for each generated node $v \in \mathcal{V}^+$ and select its K_{init} most semantically similar original nodes:

$$\mathcal{N}_{\mathrm{sem}}(v) = \underset{u\in\mathcal{V}}{\arg\operatorname{top}^{K_{\mathrm{init}}}} \cos(\boldsymbol{h}_v, \boldsymbol{h}_u), \qquad \mathcal{G}_{\mathrm{prov}} = \left(\mathcal{V} \cup \mathcal{V}^+, \mathcal{E} \cup \mathcal{E}_{\mathrm{init}}\right), \tag{13}$$

where $\mathcal{E}_{\mathrm{init}} = \{(v,u) \mid v \in \mathcal{V}^+, u \in \mathcal{N}_{\mathrm{sem}}(v)\}$ provides temporary structural context. We apply the frozen shadow model F_{sh} to $\mathcal{G}_{\mathrm{prov}}$ to obtain provisional posteriors $\boldsymbol{\rho}_{i,\mathrm{prov}}$ and confidence trends $\boldsymbol{\chi}_{i,\mathrm{prov}}$ for all nodes. These quantities are then used to construct $\boldsymbol{a}_{vu,\mathrm{prov}}$, with structural confidence score $s_{vu}^{\mathrm{inv}} = g_\psi(\boldsymbol{a}_{vu,\mathrm{prov}})$. For each augmented node v, we construct the recovered edge set $\mathcal{E}^+$ as:

$$\mathcal{E}^+ = \left\{ (v,u) \;\middle|\; v \in \mathcal{V}^+,\ s_{vu}^{\mathrm{inv}} \geq \tau_e,\ u \in \underset{w\in\mathcal{C}_v^{\mathrm{edge}}}{\arg\operatorname{top}^{K_{\mathrm{edge}}}} s_{vw}^{\mathrm{inv}} \right\}, \tag{14}$$

where τ_e is the score threshold and K_{edge} is the maximum recovered degree. To remove structurally unsupported augmented nodes, we further impose a minimum recovered degree constraint:

$$\mathcal{V}^+ \leftarrow \left\{v \in \mathcal{V}^+ \;\middle|\; \deg^+(v) \geq K_{\min}\right\}, \qquad \deg^+(v) = \left|\left\{u : (v,u) \in \mathcal{E}^+\right\}\right|. \tag{15}$$

Here, $K_{\min}$ serves as the minimum structural-support threshold for retaining an augmented node. Hence, the resulting augmented graph is $\mathcal{G}^{\mathrm{aug}} = (\mathcal{V} \cup \mathcal{V}^+, \mathcal{E} \cup \mathcal{E}^+)$.

3.5 Multi-task Learning from Hierarchical Supervision

To preserve the fine-grained semantics introduced by auxiliary subclasses while maintaining consistency with the original label space, we formulate hierarchical supervision as a multi-task learning framework. Let $\mathcal{Y}$ denote the original C classes and $\mathcal{A} = \bigcup_{c_t} \mathcal{A}_{c_t}$ denote the auxiliary subclasses, where $\mathrm{par}(j) = c_t$ for $j \in \mathcal{A}_{c_t}$. To suppress parent–child competition, we adapt the silencing strength to sample-wise semantic ambiguity. Specifically, we define a label-dependent mask as:

$$M_{y,k} = \mathbf{1}[y \in \mathcal{Y},\, \mathrm{par}(k) = y \;\vee\; y \in \mathcal{A},\, k = \mathrm{par}(y)]\,, \tag{16}$$

where $y \in \mathcal{Y} \cup \mathcal{A}$ denotes the ground-truth label and $k \in \mathcal{Y} \cup \mathcal{A}$ indexes a candidate output label. Let $\boldsymbol{z}_i \in \mathbb{R}^{C+|\mathcal{A}|}$ denote the logits produced by the final classifier. We then adjust the logits as:

$$\widetilde{\boldsymbol{z}}_i = \boldsymbol{z}_i + \log \boldsymbol{\nu} + \log \lambda_s \, \boldsymbol{M}_{y_i}, \qquad 0 < \lambda_s < 1, \tag{17}$$

where $\boldsymbol{\nu}$ denotes label frequencies over $\mathcal{V}_{\mathrm{tr}} \cup \mathcal{V}^+$. The fine-grained subvlass-level objective is:

$$\mathcal{L}_{\mathrm{leaf}} = -\frac{1}{|\mathcal{V}_{\mathrm{tr}} \cup \mathcal{V}^+|} \sum_{i \in \mathcal{V}_{\mathrm{tr}} \cup \mathcal{V}^+} \log \frac{\exp(\widetilde{z}_{i,y_i})}{\sum_{k \in \mathcal{Y} \cup \mathcal{A}} \exp(\widetilde{z}_{i,k})}. \tag{18}$$

To further align the learned subclass semantics with the original classification task, we aggregate each auxiliary subclass into its parent class via:

$$z_{i,c}^{\mathrm{par}} = \log\Big(\exp(z_{i,c}) + \textstyle\sum_{j \in \mathcal{A}:\mathrm{par}(j)=c} \exp(z_{i,j})\Big)\,, \quad c \in \mathcal{Y}. \tag{19}$$

The corresponding parent-level objective is:

$$\mathcal{L}_{\mathrm{par}} = -\frac{1}{|\mathcal{V}_{\mathrm{tr}} \cup \mathcal{V}^+|} \sum_{i \in \mathcal{V}_{\mathrm{tr}} \cup \mathcal{V}^+} \log \frac{\exp\big(z_{i,\bar{y}_i}^{\mathrm{par}}\big)}{\sum_{c \in \mathcal{Y}} \exp\big(z_{i,c}^{\mathrm{par}}\big)}, \tag{20}$$

where $\bar{y}_i = y_i$ if $y_i \in \mathcal{Y}$, and $\bar{y}_i = \mathrm{par}(y_i)$ if $y_i \in \mathcal{A}$. The two tasks are jointly optimized as:

$$\mathcal{L} = \mathcal{L}_{\mathrm{leaf}} + \lambda_{\mathrm{par}} \mathcal{L}_{\mathrm{par}}. \tag{21}$$

This multi-task objective learns fine-grained auxiliary semantics, enabling auxiliary categories to enrich representation learning without altering the final label space. During inference, auxiliary categories are mapped back to $\mathcal{Y}$, with predictions obtained as $\widehat{y}_i = \mathrm{argmax}_{c \in \mathcal{Y}} \, z_{i,c}^{\mathrm{par}}$.

3.6 Theoretical Guarantees

In this part, we provide a theoretical analysis of **BEAST**. Proofs are given in Appendix A. Fix a tail class c_t and its competitor set $\mathcal{C}_{c_t}$. Let $\boldsymbol{u} = \boldsymbol{h}/\|\boldsymbol{h}\|_2$ be the normalized representation of a candidate, and let the prototypes $\boldsymbol{\mu}_c$ have unit norm. We write Δ for the margin of Equation 6 at this candidate and Δ' for its value after $\boldsymbol{u}$ and the prototypes are perturbed.

Theorem 3.1. *Suppose the perturbed unit vectors satisfy*

$$\|\boldsymbol{u}' - \boldsymbol{u}\|_2 \le \epsilon_h, \qquad \|\boldsymbol{\mu}_c' - \boldsymbol{\mu}_c\|_2 \le \epsilon_\mu \quad \textit{for } c \in \{c_t\} \cup \mathcal{C}_{c_t}.$$

Then

$$|\Delta' - \Delta| \le 2(\epsilon_h + \epsilon_\mu). \tag{22}$$

If $\Delta > 2(\epsilon_h + \epsilon_\mu)$, the candidate stays on the target side of every selected competitor boundary.

Theorem 3.1 bounds the change of the boundary margin by the representation error and the prototype error. A larger margin therefore certifies tolerance to larger perturbations.

We next analyze the edges recovered by Equation 14. We count each undirected pair once, require $K_{\min} \ge 1$, and delete incident edges whenever an augmented node is discarded. Let $\mathscr{F}$ contain all the information that determines the selection, and let M_{vu} record the edge state of a candidate pair in a reference random augmented graph,

$$M_{vu} \in \{0, 1\}, \qquad p_{vu} := \mathbb{P}(M_{vu} = 1 \mid \mathscr{F}).$$

Whenever $\mathcal{E}^+$ is nonempty, the fraction of wrong edges among those we keep is

$$\mathrm{FEP}(\mathcal{E}^+) = \frac{1}{|\mathcal{E}^+|} \sum_{(v,u) \in \mathcal{E}^+} (1 - M_{vu}), \tag{23}$$

and we set it to zero when no edge is recovered.

Table 1: Performance comparison on Cora, PubMed, and Citeseer across four evaluation metrics. All results are averaged over five independent runs with different random seeds, and the corresponding standard deviations are reported. All methods are evaluated using identical data splits and experimental settings. We mark the best in **bold** and runners-up with underline.

Method	Cora				Pubmed				Citeseer			
	Acc	F1	GMean	bAcc	Acc	F1	GMean	bAcc	Acc	F1	GMean	bAcc
SMOTE	71.8±2.7	70.8±2.1	82.7±1.7	71.8±2.7	68.5±6.2	65.5±9.9	75.9±4.8	68.5±6.2	50.7±3.0	47.9±3.1	67.6±2.2	50.7±3.0
Oversampling	72.9±1.3	72.0±0.9	83.4±0.9	72.9±1.3	58.5±6.3	50.4±6.5	68.1±5.1	58.5±6.3	52.0±4.9	49.6±4.7	68.5±3.6	52.0±4.9
EmbedSMOTE	70.0±1.6	69.8±1.4	81.5±1.0	70.0±1.6	63.4±7.6	60.3±9.7	72.0±5.9	63.4±7.6	47.5±2.0	45.4±2.5	65.2±1.5	47.5±2.0
GraphSMOTE$_T$	74.4±4.0	74.2±4.3	84.4±2.6	74.4±4.0	67.9±5.0	67.8±5.0	75.5±3.9	67.9±5.0	50.5±2.8	48.1±3.2	67.4±2.1	50.5±2.8
GraphSMOTE$_O$	74.3±3.7	73.8±4.9	84.3±3.0	74.3±3.7	69.7±5.0	69.9±4.8	76.9±3.9	69.7±5.0	52.0±1.6	49.8±1.8	68.6±1.2	52.0±1.6
MixupForGraph	66.4±2.7	63.3±2.7	80.1±1.6	68.0±2.5	69.6±0.8	68.7±1.1	75.9±1.1	68.7±1.6	63.2±1.9	59.4±1.9	75.1±1.8	60.9±2.5
HierTail	72.8±5.3	73.3±4.7	83.3±3.4	72.8±5.3	72.4±8.2	72.4±7.7	79.0±6.3	72.5±8.0	49.6±9.6	44.5±9.3	66.5±7.4	49.6±9.6
LTE4G	74.2±3.8	74.0±3.8	84.3±2.4	74.2±3.8	72.9±1.5	72.6±1.9	79.4±1.2	72.9±1.5	51.4±3.8	49.2±3.7	68.1±2.8	51.4±3.8
SAVE-TAG$_O$	72.1±0.1	70.9±0.1	84.5±0.3	74.8±0.4	72.8±0.3	73.0±0.4	80.2±0.2	75.0±0.2	69.8±0.2	66.8±0.2	79.9±0.1	67.8±0.2
SAVE-TAG$_M$	73.0±0.4	71.8±0.3	85.1±0.2	75.8±0.4	71.3±0.2	71.2±0.3	79.1±0.1	73.6±0.3	69.5±0.3	65.9±0.4	79.3±0.3	66.9±0.5
SAVE-TAG$_S$	76.4±0.3	74.1±0.3	85.8±0.2	76.6±0.3	76.3±0.7	76.4±0.8	82.5±0.5	77.6±0.7	69.9±0.3	66.9±0.3	80.1±0.3	68.1±0.4
BEAST	**77.6**±0.5	**75.3**±0.3	**86.1**±0.4	**76.8**±0.4	**76.9**±0.2	**76.7**±0.1	**82.6**±0.8	**77.8**±0.6	**72.2**±0.3	**68.9**±0.7	**80.4**±0.2	**68.5**±0.3

Theorem 3.2. *Assume* $0 \le \tau_e \le 1$ *and that, for some* $\epsilon_e \ge 0$*, every candidate pair satisfies*

$$|s_{vu}^{\text{inv}} - p_{vu}| \le \epsilon_e.$$

Then

$$\mathbb{E}\big[\text{FEP}(\mathcal{E}^+) \mid \mathscr{F}\big] \le 1 - \tau_e + \epsilon_e. \tag{24}$$

The bound also holds at each augmented node we keep and requires no edge independence.

Theorem 3.2 bounds the expected fraction of wrong edges by the threshold and the score error, and the later steps preserve this bound because they only remove edges that already satisfy the threshold.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on six widely used text-attributed graph benchmarks, including three citation networks, Cora (McCallum et al., 2000), PubMed (Sen et al., 2008), and Citeseer (Giles et al., 1998), and three Amazon co-purchase graphs, Photo, Computer, and Children (McCallum et al., 2000). The citation networks serve as standard benchmarks for text-attributed node classification, whereas the Amazon datasets evaluate the scalability and generalization ability. Following established long-tailed graph learning protocols (Wang et al., 2024; Fu et al., 2023), we construct imbalanced training sets by sampling 20 labeled nodes for each head class and reducing the number of labeled nodes in tail classes according to a predefined *imbalance ratio*.

Baselines. We compare **BEAST** with representative baselines from three categories. *Data Augmentation Methods* include Oversampling (Chawla, 2003), SMOTE (Chawla et al., 2002), EmbedSMOTE (Ando & Huang, 2017), GraphSMOTE (Zhao et al., 2021), and MixupForGraph (Wang et al., 2021), which perform sample generation through replication, interpolation, or graph-aware augmentation. *Long-Tailed Graph Learning Methods* include LTE4G (Yun et al., 2022) and HierTail (Wang et al., 2024), which address class imbalance through frequency-aware modeling. *LLM-based Semantic Augmentation Method* includes SAVE-TAG (Wang et al., 2026b), which leverages LLM-based text augmentation and confidence-aware topology assignment.

Implementation Details. We implement our **BEAST** using PyTorch and PyTorch Geometric. Following prior long-tailed graph learning settings (Wang et al., 2024; 2026b), we construct long-tailed training sets with 20 labeled nodes per head class and $20 \times IR$ nodes per tail class, where IR denotes the imbalance ratio and is set to 0.2 by default. Both node synthesis and shadow GNN training use only training nodes. For textual representation, we use Sentence-BERT (Reimers & Gurevych, 2019) as the frozen encoder and Llama3-8B-Instruct (Grattafiori et al., 2024) as the default LLM for node synthesis. GCN (Kipf & Welling, 2016) is adopted as the backbone, with all trainable components optimized using Adam. We evaluate accuracy (Acc), macro-F1 (F1), balanced accuracy (bAcc), and geometric mean (GMean). All experiments use the same data splits, with results reported as the mean and standard deviation over five runs conducted on an NVIDIA A100 GPU.

Table 2: Performance comparison on Photo, Computer, and Children across four evaluation metrics. Following the same protocol, best results are shown in **bold**, and second-best results are underlined.

Method	Photo				Computer				Children			
	Acc	F1	GMean	bAcc	Acc	F1	GMean	bAcc	Acc	F1	GMean	bAcc
SMOTE	51.8±1.5	54.0±1.2	82.6±0.6	71.2±0.9	47.5±1.0	43.6±0.4	78.5±0.5	65.4±0.8	18.5±0.3	18.8±0.5	60.5±0.2	37.9±0.2
Oversampling	52.6±1.5	54.7±1.1	83.1±0.5	72.1±0.8	47.8±1.6	43.6±0.9	78.7±0.7	65.7±1.0	18.6±0.9	19.0±0.8	60.7±0.5	38.2±0.6
EmbedSMOTE	42.6±0.9	47.3±1.3	78.9±0.8	65.5±1.3	44.7±1.7	41.3±1.4	76.8±0.8	62.8±1.1	15.9±0.4	15.1±0.4	58.1±0.7	35.0±0.8
GraphSMOTE$_T$	34.6±2.0	39.1±1.7	71.9±1.3	54.9±1.9	30.3±2.9	29.2±2.3	69.4±1.8	52.0±2.5	19.3±3.9	19.9±3.6	43.5±9.9	21.1±8.7
GraphSMOTE$_O$	34.9±1.6	39.4±1.3	72.3±1.0	55.4±1.6	31.5±3.6	30.3±2.9	70.1±2.2	53.1±3.2	19.2±3.8	17.8±3.6	43.2±9.8	20.9±8.6
MixupForGraph	43.7±2.0	50.2±2.2	81.8±1.4	70.5±2.3	49.9±4.2	45.7±3.6	79.9±1.8	67.5±2.7	22.2±1.4	20.9±1.0	51.7±2.4	35.0±1.1
HierTail	55.4±1.8	56.7±1.1	83.6±0.7	73.5±1.1	50.7±1.5	46.1±1.0	80.0±0.8	67.5±1.3	24.1±0.2	18.9±0.5	59.0±0.6	38.8±0.7
LTE4G	40.6±4.0	43.4±3.4	72.1±2.5	54.9±3.6	38.2±3.3	36.6±3.7	71.7±3.1	55.3±4.5	24.9±1.1	19.4±1.0	61.4±1.7	39.7±1.5
SAVE-TAG$_O$	54.4±1.1	56.2±1.0	84.0±0.4	73.5±0.6	61.6±1.7	53.2±1.1	83.4±0.4	**72.7**±0.6	23.8±1.5	22.5±0.5	63.9±0.3	42.3±0.4
SAVE-TAG$_M$	56.4±0.7	57.7±0.5	84.1±0.2	73.6±0.4	58.6±1.6	51.7±1.2	82.7±0.7	71.7±1.0	26.5±1.2	**23.9**±1.0	63.8±0.5	42.0±0.7
SAVE-TAG$_S$	56.1±0.6	57.9±0.4	84.4±0.2	74.2±0.3	60.4±1.6	52.4±1.0	83.1±0.7	72.2±1.1	25.4±1.2	20.3±1.0	64.0±0.3	42.4±0.4
BEAST	**66.9**±0.6	**64.6**±0.5	**84.6**±0.6	**74.8**±0.7	**64.2**±2.1	**53.3**±0.6	**84.1**±1.6	72.4±2.4	**27.0**±0.8	23.8±1.3	**64.6**±0.7	**42.8**±0.3

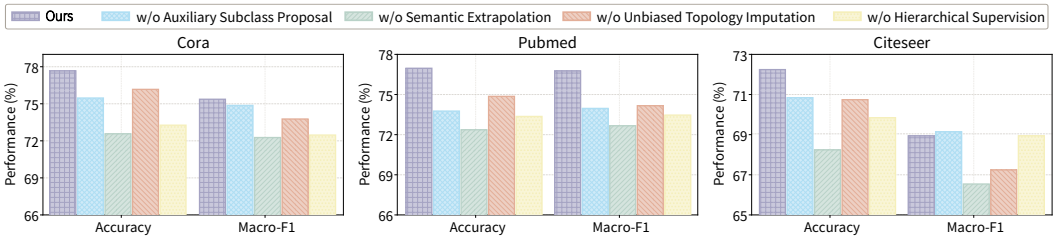


Figure 3: Ablation studies on Cora, PubMed, and Citeseer using Accuracy and Macro-F1. 'w/o' indicates the removal of a specific component. Results are averaged over five independent runs.

4.2 Performance Comparison

We conduct comprehensive experiments on six long-tailed TAG benchmarks with results summarized in Table 1 and Table 2. We can draw some key observations as follows: Obs.❶ BEAST consistently achieves the state-of-the-art performance across all the datasets except for two metrics. Notably, BEAST improves Accuracy on Citeseer from 69.9% to 72.2% over SAVE-TAG. Obs.❷ The consistent gains in Macro-F1, GMean and bAcc demonstrate that BEAST improves minority-class recognition and alleviates the prediction bias toward head classes. Obs.❸ BEAST maintains strong performance on larger-scale graphs, demonstrating its effectiveness under more complex graph structures. We attribute this to auxiliary subclass proposal with semantic extrapolation and unbiased topology imputation, which improve tail-class separability and structural reliability, while hierarchical supervision preserves fine-grained semantic diversity beyond coarse tail labels.

4.3 Ablation Study

We conduct ablation studies with four variants by removing Auxiliary Subclass Proposal, Semantic Extrapolation, Topology Imputation, and Hierarchical Supervision, respectively. Results are shown in Figure 3, from which we make the following observations. **Obs.❶** Removing any component generally degrades performance, confirming that each component provides complementary contributions to the overall framework. **Obs.❷** Semantic extrapolation causes the largest drop when removed, highlighting its key role in enriching tail-class boundaries by providing discriminative semantic evidence for underrepresented classes. **Obs.❸** Auxiliary subclass proposal, topology imputation, and hierarchical supervision provide complementary semantic and structural benefits, where semantic guidance improves class representation, topology completion restores message-passing information, and hierarchical supervision maintains consistency between auxiliary and original labels.

4.4 In-Depth Analysis

Effectiveness of LLM-Guided Semantic Extrapolation. Table 3 examines whether our semantic augmentation provides supervision before structural modeling. Conventional zero-shot and few-shot generation mainly increase sample quantity, while LLM-based extrapolation imposes semantic constraints. In contrast, our BEAST explores diverse semantics to generate samples toward ambiguous regions, enriching tail-class discrimination. The consistent gains across benchmarks demonstrate that semantic extrapolation itself contributes to the improvement beyond structural modeling.

Visualization of Semantic Boundary Enrichment. Figure 4 (a) visualizes the distributions of original and generated samples. SMOTE mainly produces samples within locally observed regions,

Table 3: Comparison of different semantic extrapolation strategies on Cora, PubMed, and Citeseer before structural modeling. Results are averaged over five independent runs, with standard deviations reported. Best results are shown in **bold**, and second-best results are underlined.

Method	Cora				Pubmed				Citeseer			
	Acc	F1	GMean	bAcc	Acc	F1	GMean	bAcc	Acc	F1	GMean	bAcc
Original	72.5±1.4	70.4±1.1	84.5±0.6	74.7±0.9	55.1±8.2	47.8±9.8	62.2±7.9	51.5±9.7	53.7±0.1	49.6±2.2	71.4±0.9	55.9±1.3
Zero-shot	68.3±0.5	63.3±0.7	81.1±0.4	69.4±0.7	71.4±0.4	71.4±0.5	78.3±0.4	72.3±0.5	64.8±0.5	62.1±0.5	78.0±0.3	65.2±0.5
Few-shot	73.9±0.6	72.7±0.6	85.4±0.2	76.1±0.3	72.0±0.3	71.6±0.3	79.2±0.2	73.6±0.3	68.2±0.9	65.6±0.7	79.8±0.4	67.8±0.5
LLM_O	73.7±0.7	72.1±0.7	85.3±0.3	76.1±0.4	71.5±0.3	71.4±0.3	78.8±0.3	73.1±0.4	69.6±0.8	67.0±0.6	**80.5**±0.4	**68.8**±0.5
LLM_M	74.2±0.4	72.6±0.3	85.6±0.1	76.5±0.2	71.9±0.4	72.0±0.4	79.0±0.2	73.4±0.3	69.5±0.6	66.8±0.4	80.3±0.2	68.6±0.3
LLM_S	74.5±0.4	72.5±0.5	85.5±0.2	76.3±0.3	72.5±0.4	72.5±0.4	79.4±0.3	73.8±0.3	69.2±0.6	66.5±0.5	80.1±0.3	68.3±0.5
Ours	**76.2**±0.2	**73.7**±0.5	**85.8**±0.2	**76.8**±0.1	**74.3**±0.3	**74.1**±0.6	**80.6**±0.2	**75.2**±0.4	**70.7**±0.6	**67.2**±0.4	80.2±0.5	68.4±0.6

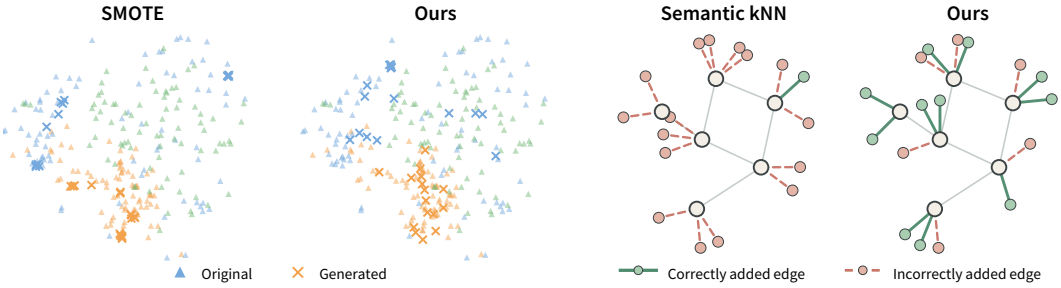


Figure 4: (a) Semantic boundary enrichment on Cora. (b) Topology completion quality on Cora, measured by correctly and incorrectly added edges.

leading to limited expansion of tail-class semantics. In contrast, **BEAST** generates samples over a broader semantic range, including sparsely observed regions close to competing classes. Such boundary-oriented extrapolation enriches the semantic evidence around ambiguous regions and provides stronger supervision for learning discriminative tail-class boundaries.

Hyper-parameter Analysis To validate the robustness of **BEAST** to different hyperparameter choices, we examine the boundary margin τ_{bd} and edge threshold τ_e in Figure 5. A moderate τ_{bd} achieves the best overall performance, indicating that properly controlling the boundary region is crucial for generating informative tail-class samples with sufficient class discrimination. Meanwhile, performance remains relatively stable across different τ_e values, suggesting that the topology refinement process is not sensitive to the exact edge filtering threshold due to the learned structural evidence. These results demonstrate that **BEAST** is robust to hyperparameter variations and maintains stable performance under different configurations.

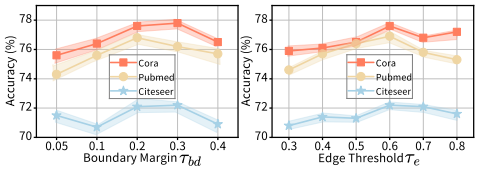


Figure 5: Hyperparameters Analysis on Cora, Pubmed, and Citeseer over five independent runs with different random seeds.

Qualitative Analysis of Topology Imputation. We evaluate **BEAST**'s topology imputation through a controlled edge-recovery study on Cora, where tail-class edges are partially masked and recovered under the same protocol. As shown in Figure 4 (b), across six low-degree cases, semantic kNN correctly recovers only 1 of 17 edges, while **BEAST** recovers 11, reducing false connections from 16 to 6. This demonstrates that auxiliary semantics alone may fail to provide reliable neighbors for message passing, whereas **BEAST** leverages structural evidence to suppress spurious edges.

5 Conclusion

In this work, we study long-tailed text-attributed graph learning from the perspectives of semantic insufficiency and structural bias. We propose **BEAST**, which introduces auxiliary subclasses to guide boundary-aware semantic extrapolation, imputes reliable topology for generated nodes from connectivity patterns learned on a rebalanced shadow graph, and exploits hierarchical multi-task supervision over subclass and parent labels. Extensive experiments on multiple benchmarks demonstrate consistent gains over representative baselines, highlighting the importance of enriching tail semantics while restoring reliable structural context through effective multi-task learning.

AI USE STATEMENT

In this work, we did not use generative AI tools to generate synthetic datasets, develop theoretical or conceptual frameworks, formulate mathematical claims or proofs, propose hypotheses, design research methodology or experiments, or interpret experimental results. In addition to our method which is required to utilize the LLM, we used generative AI tools to assist with editing the manuscript for grammar, and readability, and to generate or refine portions of software code used in our experiments. All AI-assisted text was carefully reviewed and revised by the authors to ensure technical accuracy and consistency with the intended meaning. AI-generated or AI-assisted code was manually inspected, tested, and validated by the authors before use. We take full responsibility for the final content of this work, including any text or code produced with the aid of generative AI.

REFERENCES

Shin Ando and Chun Yuan Huang. Deep over-sampling framework for classifying imbalanced data. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 770–785. Springer, 2017.

Nitesh V Chawla. C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *Proceedings of the ICML*, volume 3, pp. 1–9. CIBC Toronto, ON, Canada, 2003.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Deli Chen, Yankai Lin, Guangxiang Zhao, Xuancheng Ren, Peng Li, Jie Zhou, and Xu Sun. Topology-imbalance learning for semi-supervised node classification. *Advances in neural information processing systems*, 34:29885–29897, 2021.

Yi Fang, Dongzhe Fan, Daochen Zha, and Qiaoyu Tan. Gaugllm: Improving graph contrastive learning for text-attributed graphs with large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 747–758, 2024.

Xingcheng Fu, Yuecen Wei, Qingyun Sun, Haonan Yuan, Jia Wu, Hao Peng, and Jianxin Li. Hyperbolic geometric graph representation learning for hierarchy-imbalance node classification. In *Proceedings of the ACM Web Conference 2023*, pp. 460–468, 2023.

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*, pp. 89–98, 1998.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gaoyang Guo, Chaokun Wang, Bencheng Yan, Yunkai Lou, Hao Feng, Junchao Zhu, Jun Chen, Fei He, and Philip S Yu. Learning adaptive node embeddings across graphs. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):6028–6042, 2022.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Rodrigo Hormazabal, Changyoung Park, Soonyoung Lee, Sehui Han, Yeonsik Jo, Jaewan Lee, Ahra Jo, Seung Hwan Kim, Jaegul Choo, Moontae Lee, et al. Cede: A collection of expert-curated datasets with atom-level entity annotations for optical chemical structure recognition. In *Proceedings of the Conference on Neural Information Processing Systems*, pp. 27114–27126, 2022.

Xinyu Hu, Donghai Guan, Weiwei Yuan, Qi Zhu, and Çetin Kaya Koç. Adaptive imbalanced node classification graph contrastive learning. *Neurocomputing*, pp. 131280, 2025.

Bowen Jin, Gang Liu, Chi Han, Meng Jiang, Heng Ji, and Jiawei Han. Large language models on graphs: A comprehensive survey. *IEEE Transactions on Knowledge and Data Engineering*, 36 (12):8622–8642, 2024.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Mengran Li, Pengyu Zhang, Wenbin Xing, Yijia Zheng, Klim Zaporojets, Junzhou Chen, Ronghui Zhang, Yong Zhang, Siyuan Gong, Jia Hu, et al. A survey of large language models for data challenges in graphs. *Expert Systems with Applications*, pp. 129643, 2025.

Wen-Zhi Li, Chang-Dong Wang, Hui Xiong, and Jian-Huang Lai. Graphsha: Synthesizing harder samples for class-imbalanced node classification. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 1328–1340, 2023.

Wei Lin, Xu Peng, Zhengtao Yu, and Taisong Jin. Hypergraph neural architecture search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13837–13845, 2024.

Zemin Liu, Yuan Li, Nan Chen, Qian Wang, Bryan Hooi, and Bingsheng He. A survey of imbalanced learning on graphs: Problems, techniques, and future directions. *IEEE Transactions on Knowledge and Data Engineering*, 37(6):3132–3152, 2025.

Yihong Ma, Yijun Tian, Nuno Moniz, and Nitesh V Chawla. Class-imbalanced learning on graphs: A survey. *ACM Computing Surveys*, 57(8):1–16, 2025.

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.

Zhihong Pan, Zhijie Zhong, Weisheng Li, Junming Zhou, Shupeng Li, Ronghua Lin, and Yong Tang. Unis2a: Unified semantic and structural augmentation for text-attributed graphs with large language models. *Neurocomputing*, pp. 133800, 2026.

Joonhyung Park, Jaeyun Song, and Eunho Yang. Graphens: Neighbor-aware ego network synthesis for class-imbalanced node classification. In *International conference on learning representations*, 2021.

Jiawen Qin, Qingyun Sun, Lyujin Xu, Cheng Ji, and Jianxin Li. Immix: Class-imbalanced node classification via prototypical selective mixup augmentation. *Pattern Recognition*, pp. 112749, 2025.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–106, 2008.

Jaeyun Song, Joonhyung Park, and Eunho Yang. Tam: topology-aware margin loss for class-imbalanced node classification. In *International conference on machine learning*, pp. 20369–20383. PMLR, 2022.

Guangxin Su, Hanchen Wang, Jianwei Wang, Wenjie Zhang, Ying Zhang, and Jian Pei. Large language models meet text-attributed graphs: A survey of integration frameworks and applications. *arXiv preprint arXiv:2510.21131*, 2025.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Changhai Wang, Zhe Huang, Yuwei Xu, and Yaoli Xu. Imbalanced node classification on graphs with graph neural networks: A survey. *Electronic Research Archive*, 34(3):1742–1784, 2026a.

Haohui Wang, Baoyu Jing, Kaize Ding, Yada Zhu, Wei Cheng, Si Zhang, Yonghui Fan, Liqing Zhang, and Dawei Zhou. Mastering long-tail complexity on graphs: Characterization, learning, and generalization. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3045–3056, 2024.

Haoyu Peter Wang, Shikun Liu, Rongzhe Wei, and Pan Li. Generalization principles for inference over text-attributed graphs with large language models. In *Forty-second International Conference on Machine Learning*, 2025a.

Leyao Wang, Yu Wang, Bo Ni, Yuying Zhao, Hanyu Wang, Yao Ma, and Tyler Derr. Save-tag: Llm-based interpolation for long-tailed text-attributed graphs. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1447–1458, 2026b.

Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Mixup for node and graph classification. In *Proceedings of the web conference 2021*, pp. 3663–3674, 2021.

Zehong Wang, Sidney Liu, Zheyuan Zhang, Tianyi Ma, Chuxu Zhang, and Yanfang Ye. Can llms convert graphs to text-attributed graphs? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1412–1432, 2025b.

Chuhan Wu, Fangzhao Wu, Tao Qi, Suyu Ge, Yongfeng Huang, and Xing Xie. Reviews meet graphs: Enhancing user and item representations for recommendation with hierarchical attentive graph neural network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4884–4893, 2019.

Lirong Wu, Jun Xia, Zhangyang Gao, Haitao Lin, Cheng Tan, and Stan Z Li. Graphmixup: Improving class-imbalanced node classification by reinforcement mixup and self-supervised context prediction. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 519–535. Springer, 2022.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Hao Yan, Chaozhuo Li, Ruosong Long, Chao Yan, Jianan Zhao, Wenwen Zhuang, Jun Yin, Peiyan Zhang, Weihao Han, Hao Sun, et al. A comprehensive study on text-attributed graphs: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 36:17238–17264, 2023.

Haoran Yang, Xiangyu Zhao, Sirui Huang, Qing Li, and Guandong Xu. Latex-gcl: Large language models (llms)-based data augmentation for text-attributed graph contrastive learning. *arXiv preprint arXiv:2409.01145*, 2024.

Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 974–983, 2018.

Sukwon Yun, Kibum Kim, Kanghoon Yoon, and Chanyoung Park. Lte4g: Long-tail experts for graph neural networks. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 2434–2443, 2022.

Delvin Ce Zhang, Menglin Yang, Rex Ying, and Hady W Lauw. Text-attributed graph representation learning: Methods, applications, and challenges. In *Companion Proceedings of the ACM Web Conference 2024*, pp. 1298–1301, 2024.

Jiahao Zhang, Yilong Wang, Zhiwei Zhang, Xiaorui Liu, and Suhang Wang. Unlearning inversion attacks for graph neural networks. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pp. 934–945, 2026.

Yanfu Zhang, Shangqian Gao, Jian Pei, and Heng Huang. Improving social network embedding via new second-order continuous graph neural networks. In *Proceedings of the International ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2515–2523, 2022.

Huanjing Zhao, Beining Yang, Yukuo Cen, Junyu Ren, Chenhui Zhang, Yuxiao Dong, Evgeny Kharlamov, Shu Zhao, and Jie Tang. Pre-training and prompting for few-shot node classification on text-attributed graphs. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4467–4478, 2024.

Jianan Zhao, Meng Qu, Chaozhuo Li, Hao Yan, Qian Liu, Rui Li, Xing Xie, and Jian Tang. Learning on large-scale text-attributed graphs via variational inference. *arXiv preprint arXiv:2210.14709*, 2022.

Shizhen Zhao, Xin Wen, Jiahui Liu, Chuofan Ma, Chunfeng Yuan, and Xiaojuan Qi. Learning from neighbors: Category extrapolation for long-tail learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30483–30492. IEEE, 2025.

Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 833–841, 2021.

Chuang Zhou, Jiahe Du, Huachi Zhou, Hao Chen, Feiran Huang, and Xiao Huang. Text-attributed graph learning with coupled augmentations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 10865–10876, 2025.

Mengting Zhou and Zhiguo Gong. Graphsr: A data augmentation algorithm for imbalanced node classification. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 37, pp. 4954–4962, 2023.

A Proof of Theoretical Results

A.1 Proof of the Margin Stability Bound

Proof of Theorem 3.1. For a unit vector $\boldsymbol{u}$ and unit prototypes,

$$\begin{aligned}\Delta &= d_{\mathcal{C}}(\boldsymbol{u}) - d_{c_t}(\boldsymbol{u}) \\ &= \Big(1 - \max_{c\in\mathcal{C}_{c_t}} \boldsymbol{u}^\top \boldsymbol{\mu}_c\Big) - \big(1 - \boldsymbol{u}^\top \boldsymbol{\mu}_{c_t}\big) \\ &= \min_{c\in\mathcal{C}_{c_t}} \boldsymbol{u}^\top(\boldsymbol{\mu}_{c_t} - \boldsymbol{\mu}_c) =: \min_{c\in\mathcal{C}_{c_t}} a_c,\end{aligned} \tag{25}$$

and likewise $\Delta' = \min_c a'_c$ with $a'_c = \boldsymbol{u}'^\top(\boldsymbol{\mu}'_{c_t} - \boldsymbol{\mu}'_c)$. Inserting $\boldsymbol{u}'^\top(\boldsymbol{\mu}_{c_t} - \boldsymbol{\mu}_c)$ gives

$$a'_c - a_c = (\boldsymbol{u}' - \boldsymbol{u})^\top(\boldsymbol{\mu}_{c_t} - \boldsymbol{\mu}_c) + \boldsymbol{u}'^\top\big[(\boldsymbol{\mu}'_{c_t} - \boldsymbol{\mu}_{c_t}) - (\boldsymbol{\mu}'_c - \boldsymbol{\mu}_c)\big], \tag{26}$$

so Cauchy–Schwarz, $\|\boldsymbol{u}'\|_2 = 1$ and the triangle inequality yield

$$\begin{aligned}|a'_c - a_c| &\le \epsilon_h\|\boldsymbol{\mu}_{c_t} - \boldsymbol{\mu}_c\|_2 + \|\boldsymbol{\mu}'_{c_t} - \boldsymbol{\mu}_{c_t}\|_2 + \|\boldsymbol{\mu}'_c - \boldsymbol{\mu}_c\|_2 \\ &\le 2\epsilon_h + 2\epsilon_\mu =: b \qquad \text{for all } c \in \mathcal{C}_{c_t}.\end{aligned} \tag{27}$$

Taking minima over the finite set $\mathcal{C}_{c_t}$,

$$\Delta - b = \min_c(a_c - b) \le \min_c a'_c = \Delta' \le \min_c(a_c + b) = \Delta + b, \tag{28}$$

which is $|\Delta' - \Delta| \le 2(\epsilon_h + \epsilon_\mu)$. If $\Delta > b$, then $\Delta' \ge \Delta - b > 0$, hence $a'_c > 0$ for every $c \in \mathcal{C}_{c_t}$. □

A.2 Proof of the Bound on Recovered Edges

Proof of Theorem 3.2. Conditioning on $\mathscr{F}$ fixes the scores, the set $\mathcal{E}^+$ and $N_+ = |\mathcal{E}^+|$, leaving only the indicators M_{vu} random. For $N_+ > 0$ and $(v,u) \in \mathcal{E}^+$, where $s^{\text{inv}}_{vu} \ge \tau_e$,

$$p_{vu} \ge \max\{0, s^{\text{inv}}_{vu} - \epsilon_e\} \ge \max\{0, \tau_e - \epsilon_e\} =: \pi. \tag{29}$$

Linearity of conditional expectation then gives

$$\begin{aligned}\mathbb{E}\big[\mathrm{FEP}(\mathcal{E}^+) \mid \mathscr{F}\big] &= \frac{1}{N_+}\sum_{(v,u)\in\mathcal{E}^+}(1 - p_{vu}) \\ &\le \frac{1}{N_+}\sum_{(v,u)\in\mathcal{E}^+}(1-\pi) = 1 - \pi \le 1 - \tau_e + \epsilon_e,\end{aligned} \tag{30}$$

where no joint law of the M_{vu} is used. For $N_+ = 0$,

$$\mathrm{FEP}(\mathcal{E}^+) = 0 \le 1 - \tau_e \le 1 - \tau_e + \epsilon_e. \tag{31}$$

For an augmented node v that is kept,

$$\mathcal{E}^+(v) = \{(v, u) \in \mathcal{E}^+\}, \qquad |\mathcal{E}^+(v)| \ge K_{\min} \ge 1, \tag{32}$$

and each of its pairs satisfies $s_{vu}^{\mathrm{inv}} \ge \tau_e$, so the same conditional expectation argument applies to $\mathcal{E}^+(v)$ and proves the bound at each node. □

B Further Implementation Details

B.1 Datasets Details

We provide additional details of the six text-attributed graph benchmarks used in our experiments below, with their statistics summarized in Table 4. **Cora** is a citation network with 2,708 papers and 10,858 citation links. Each paper is associated with textual content and belongs to one of seven research categories. **PubMed** contains 19,717 biomedical publications and 88,670 citation links. The papers are grouped into three diabetes-related categories with textual attributes describing their content. **Citeseer** consists of 3,186 scientific publications and 8,450 citation links. Each paper is associated with textual information and assigned to one of six research categories. **Photo** is an Amazon co-purchase graph containing 48,362 products and 500,939 edges. Product descriptions serve as textual attributes, and the nodes are divided into 12 categories. **Computer** contains 87,229 products connected by 721,081 co-purchase relations. Each product is associated with textual information and belongs to one of 10 categories. **Children** consists of 76,875 product nodes and 1,554,578 co-purchase edges. It contains 24 product categories and is the largest one in terms of edge count.

Table 4: Statistics of the datasets used in our experiments. "#Tail" denotes the number of tail classes under the adopted long-tailed setting.

Name	#Nodes	#Edges	#Class	#Tail
Cora	2,708	10,858	7	5
PubMed	19,717	88,670	3	2
Citeseer	3,186	8,450	6	4
Photo	48,362	500,939	12	8
Computer	87,229	721,081	10	6
Children	76,875	1,554,578	24	15

B.2 Baselines Details

We provide additional details of the representative baselines considered in our experiments, covering data augmentation, long-tailed graph learning, and LLM-based augmentation methods.

- **Oversampling** (Chawla, 2003): directly replicates minority-class samples to alleviate class imbalance.
- **SMOTE** (Chawla et al., 2002): synthesizes minority samples through interpolation between neighboring instances in the input space.
- **EmbedSMOTE** (Ando & Huang, 2017): performs interpolation in the learned embedding space to generate minority representations.
- **GraphSMOTE** (Zhao et al., 2021): extends SMOTE to graph data by synthesizing minority node representations while considering graph structure.
- **MixupForGraph** (Wang et al., 2021): applies mixup-based interpolation to graph representations for minority-class augmentation.
- **LTE4G** (Yun et al., 2022): employs specialized expert models for different class-frequency groups in long-tailed graph learning.
- **HierTail** (Wang et al., 2024): exploits hierarchical class relations to enhance the learning of tail classes.
- **SAVE-TAG** (Wang et al., 2026b): leverages LLM-based textual interpolation and confidence-aware edge assignment for synthesized nodes.

Overall, these baselines cover representative augmentation, imbalance-aware learning, and LLM-based strategies, providing a comprehensive comparison with our method.

B.3 Evaluation Metrics

We evaluate node classification using Accuracy (Acc), Macro-F1 (F1), balanced accuracy (bAcc), and geometric mean (GMean). Let C denote the number of classes and N the number of test nodes. For each class c, we adopt a one-vs-rest view and denote the corresponding numbers of true positives, true negatives, false positives, and false negatives by TP_c, TN_c, FP_c, and FN_c, respectively. The class-wise recall, precision, specificity, and F1 score are defined as:

$$\begin{aligned} \mathrm{Rec}_c &= \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}, \quad \mathrm{Prec}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c}, \\ \mathrm{Spec}_c &= \frac{\mathrm{TN}_c}{\mathrm{TN}_c + \mathrm{FP}_c}, \quad \mathrm{F1}_c = \frac{2\,\mathrm{Prec}_c\,\mathrm{Rec}_c}{\mathrm{Prec}_c + \mathrm{Rec}_c}. \end{aligned} \tag{33}$$

Accuracy, Macro-F1, and balanced accuracy are then computed as:

$$\mathrm{Acc} = \frac{\sum_{c=1}^{C} \mathrm{TP}_c}{N}, \qquad \mathrm{F1} = \frac{1}{C}\sum_{c=1}^{C} \mathrm{F1}_c, \qquad \mathrm{bAcc} = \frac{1}{C}\sum_{c=1}^{C} \mathrm{Rec}_c. \tag{34}$$

Following the one-vs-rest multi-class convention, GMean jointly measures the macro-averaged sensitivity and specificity:

$$\mathrm{GMean} = \sqrt{\left(\frac{1}{C}\sum_{c=1}^{C} \mathrm{Rec}_c\right)\left(\frac{1}{C}\sum_{c=1}^{C} \mathrm{Spec}_c\right)}. \tag{35}$$

All four metrics are reported as percentages. Higher values indicate better classification performance, while F1, bAcc, and GMean are less dominated by head classes than standard accuracy under class imbalance.

B.4 Prompt Template Design

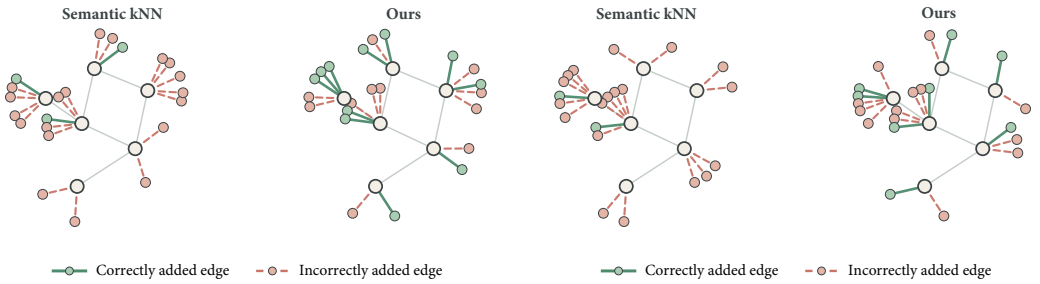
We employ structured prompts to guide the LLM throughout semantic extrapolation. Specifically, the LLM is used for auxiliary category discovery, boundary-oriented node generation, and semantic verification. For clarity and reproducibility, we provide the prompt templates below. Terms enclosed by braces denote instance-specific information that is filled dynamically during execution.

B.4.1 Prompt for auxiliary category discovery

For each tail classc_t, we provide its class name and semantic competitors identified by prototype similarity. The LLM is instructed to discover fine-grained semantic directions that remain related to the target class while being distinguishable from its competitors. As shown in Figure 5, the generated `keyword` specifies an auxiliary semantic direction, while `class_info` provides a more explicit description to reduce ambiguity. These candidates are subsequently filtered according to their distances to the target and prototypes before being used for node generation.

Table 5: Prompt template for auxiliary category discovery.

System:	You are a precise research-data assistant. Follow the requested JSON schema exactly and do not add markdown.
User:	Target tail class: {TARGET_CLASS} . Competing classes: {COMPETING_CLASSES} . Propose {NUM_CANDIDATES} scientific keywords or fine-grained auxiliary topic classes that are semantically related to the target class while remaining distinguishable from the competing classes.
Assistant:	Return a JSON array, where each item contains “keyword”, “class_info”, and “reason”.



(a) Citeseer. (b) Cora.

Figure 6: Qualitative analysis of connectivity inference on Citeseer and Pubmed.

B.4.2 Prompt for boundary-oriented node generation.

Given an accepted auxiliary category, we further prompt the LLM to instantiate it as realistic node text. As shown in Figure 6, besides the target and competitor classes, we provide representative training samples from the target class to preserve the textual style and schema of the original dataset. The generated candidates are not directly inserted into the graph. We first encode them using the same text encoder employed for the original nodes and retain candidates satisfying the semantic constraints defined in the main paper. This separates semantic generation from geometric filtering and prevents unconstrained LLM outputs from directly affecting model training.

Table 6: Prompt template for boundary-oriented node generation.

System:	You are a precise research-data assistant. Follow the requested JSON schema exactly and do not add markdown.
User:	Dataset: {DATASET}. Target tail class: {TARGET_CLASS}. Auxiliary category: {KEYWORD}. Category description: {CLASS_INFO}. Competing classes: {COMPETING_CLASSES}. Previously generated titles: {PREVIOUS_TITLES}. Generate {BATCH_SIZE} distinct academic-paper records that reflect both the target tail class and the auxiliary category, while remaining distinguishable from the competing classes. Follow the style of the reference examples but do not copy them.
Assistant:	Return a JSON array, where each item contains “title” and “abstract”. Each abstract should contain 35–65 words.

B.4.3 Prompt for LLM Critic

While geometric filtering evaluates the relative position of generated nodes in the representation space, it cannot fully determine whether the generated content is semantically valid or consistent with the intended auxiliary category. We therefore introduce an LLM-based semantic critic to perform an additional semantic verification before incorporating generated nodes into the graph. As shown in Table 7, the prompt explicitly provides the target tail class, the corresponding auxiliary category, and the generated candidate text. Based on this information, the LLM is instructed to examine whether the candidate is semantically consistent with the auxiliary category while remaining compatible with its parent tail class. Ambiguous, off-topic, or semantically inconsistent generations are rejected. The critic outputs a binary decision together with a short justification, enabling a lightweight and interpretable semantic filtering step.

Table 7: Prompt template for LLM-based semantic critic.

System:	You are a precise research-data assistant. Follow the requested JSON schema exactly and do not add markdown.
User:	Target tail class: {TARGET_CLASS}. Auxiliary category: {AUXILIARY_CATEGORY}. Competing class: {COMPETING_CLASS}. Candidate text: {CANDIDATE_TEXT}. Determine whether the candidate is semantically consistent with the auxiliary category and remains compatible with the target tail class, while avoiding ambiguity or semantic overlap with the competing class.
Assistant:	Return "pass" if the candidate satisfies the above conditions; otherwise return "reject", together with a short reason.

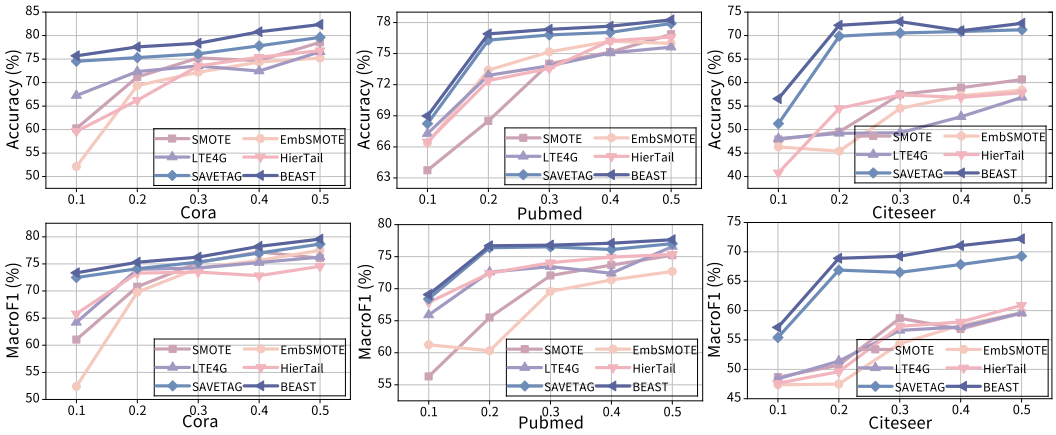


Figure 7: Performance comparison under different imbalance ratios on Cora, Pubmed, and Citeseer. Results are reported in terms of Accuracy and Macro-F1.

C Additional Experiments

C.1 Robustness to Varying Imbalance Ratio

We further evaluate the sensitivity of different methods to varying imbalance ratios on Cora, Pubmed, and Citeseer. As the imbalance ratio increases, the performance of all methods generally improves due to the increased availability of tail-class samples. Nevertheless, **BEAST** consistently achieves strong performance across different imbalance levels on both Accuracy and Macro-F1. In particular, its advantage is more pronounced under severe imbalance, especially on Cora and Citeseer, indicating that **BEAST** can better mitigate the semantic and structural deficiencies of tail classes when supervision is highly limited. As the class distribution becomes more balanced, the performance gaps gradually narrow, while **BEAST** remains consistently competitive. These results demonstrate the effectiveness of **BEAST** across different degrees of class imbalance.

C.2 Additional Qualitative Analysis of Graph Structure Restoration

As shown in Figure 6, we further compare semantic kNN with **BEAST** on Citeseer and Pubmed. Consistent with the observations on Cora reported in the main text, semantic kNN introduces a considerable number of incorrect connections, since textual similarity alone does not necessarily imply structural compatibility. In contrast, **BEAST** consistently recovers more ground-truth edges while suppressing spurious semantic neighbors across both datasets. The improvement is particularly evident around locally ambiguous regions, where semantically similar candidates may belong



(a) Citeseer. (b) Pubmed.

Figure 8: Visualization of tail-class semantic boundary enrichment on Citeseer and Cora.

to different structural contexts. These results further demonstrate that the learned connectivity inference captures structural evidence beyond pairwise semantic similarity and generalizes consistently across different TAG benchmarks.

C.3 Generalization across GNN Backbones

We further evaluate **BEAST** with different GNN backbones, including GCN (Kipf & Welling, 2016), GAT (Veličković et al., 2017), GraphSAGE (Hamilton et al., 2017), and GIN (Xu et al., 2018). As shown in Table 9, **BEAST** maintains competitive performance across different architectures, indicating that its effectiveness is not restricted to a specific GNN backbone. Among them, GCN achieves the best overall performance on all three datasets, while GIN and GAT remain close on several metrics. One possible reason is that the topology completion in **BEAST** already filters and restores informative structural connections, allowing the simple and stable neighborhood aggregation of GCN to effectively exploit the refined topology. In contrast, more complex attention or aggregation mechanisms may introduce additional modeling variability without consistently benefiting from the recovered edges, particularly under long-tailed supervision. GraphSAGE shows a more noticeable degradation, especially on Pubmed. Overall, these results demonstrate that **BEAST** generalizes well across different message-passing architectures, with GCN providing the most compatible backbone under our setting.

C.4 Additional Visualization of Boundary Enrichment

As shown in Figure 8, we further visualize the semantic augmentation on Cora and Citeseer. Consistent with the observations on PubMed, SMOTE mainly generates samples within locally observed regions, whereas **BEAST** expands tail-class semantics toward broader and more ambiguous inter-class regions. These results further demonstrate that **BEAST** consistently enriches the semantic coverage around tail-class boundaries across different datasets.

C.5 Compatibility with Different LLM Backbones

To examine the influence of LLM choice, we instantiate **BEAST** with Llama3-8B-Instruct, Qwen2.5-7B-Instruct, and Mistral-7B-Instruct. As shown in Table 8, all three backbones achieve consistently strong performance across Cora, Pubmed, and Citeseer. Llama3-8B-Instruct obtains the best overall results, while Qwen2.5-7B-Instruct and Mistral-7B-Instruct remain competitive with only moderate variations across datasets and metrics. Notably, replacing the underlying LLM does not lead to substantial performance degradation, suggesting that the effectiveness of **BEAST** mainly stems from its semantic extrapolation mechanism rather than a particular model-specific capability. These results demonstrate that **BEAST** is robust to different LLM backbones and can generalize across instruction-tuned language models.

Table 8: Performance of **BEAST** with different LLM backbones on Cora, Pubmed, and Citeseer.

Method	Cora		Pubmed		Citeseer	
	Acc	Macro-F1	Acc	Macro-F1	Acc	Macro-F1
Qwen2.5-7B-Instruct	76.8	75.2	76.2	75.7	70.9	68.7
Mistral-7B-Instruct	76.7	74.8	76.5	75.2	71.8	67.2
Llama3-8B-Instruct	**77.6**	**75.3**	**76.9**	**76.7**	**72.2**	**68.9**

Table 9: Performance comparison with different GNN backbones on Cora, Pubmed, and Citeseer.

Backbone	Cora		Pubmed		Citeseer	
	Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
GCN	**77.6±0.5**	**75.3±0.3**	**76.9±0.2**	**76.7±0.1**	**72.2±0.3**	**68.9±0.7**
GAT	76.8±0.5 ↓0.8	74.2±0.4 ↓1.1	73.9±0.6 ↓3.0	74.0±0.8 ↓2.7	71.6±0.3 ↓0.6	67.5±0.2 ↓1.4
GraphSAGE	75.2±0.3 ↓2.4	72.6±0.8 ↓2.7	69.9±0.4 ↓7.0	69.5±1.1 ↓7.2	68.7±0.7 ↓3.5	65.6±0.7 ↓3.3
GIN	77.2±0.7 ↓0.4	74.8±0.6 ↓0.5	74.9±0.2 ↓2.0	73.7±0.1 ↓3.0	70.9±0.3 ↓1.3	66.8±0.5 ↓2.1