GATTA:
Graph Active Learning with Test-Time Augmentation

Anonymous authors
Paper under double-blind review

Abstract

Test-time augmentation (TTA) has proven effective for improving model robustness and uncertainty estimation in computer vision, yet its application to graph-structured data remains largely unexplored. We introduce GATTA (Graph Active Learning with Test-Time Augmentation), a framework for enhancing active learning by aggregating predictions across multiple augmented views to produce more reliable uncertainty estimates. To address the challenge of label-preserving graph augmentations, GATTA incorporates a consistency-based filtering mechanism that discards augmented views yielding unreliable predictions.

We systematically evaluate GATTA across multiple graph datasets, GNN architectures, and acquisition strategies. Our results show that simple uncertainty-based methods, such as Entropy and Least Confidence, benefit most from TTA, achieving performance competitive with more sophisticated and computationally expensive approaches. GATTA generalizes across architectures, outperforms model-side ensemble methods such as MC Dropout. We further show that GATTA scales efficiently with both ensemble size and graph size. Extensive analysis of augmentation types, strengths, and filtering strategies provides practical guidelines for effective deployment.

Our findings demonstrate that augmenting simple methods with TTA offers a more efficient path to strong active learning performance than engineering complex acquisition functions, enabling practitioners to achieve competitive results with lower computational overhead and reduced implementation complexity.

1 Introduction

Graph neural networks (GNNs) have become the dominant paradigm for modeling relational data, achieving state-of-the-art performance across a wide range of applications, from molecular property prediction to social network analysis (Zhou et al., 2020; Wu et al., 2019b). However, most high-performance models still rely on substantial labeled data for tasks like node classification, creating a significant labeling bottleneck. This challenge is particularly acute in scientific and industrial domains where annotation requires costly domain expertise or experimental validation. (Gal et al., 2017; Litjens et al., 2017; Gilmer et al., 2017; Wu et al., 2017; Halbouni et al., 2022)

Active learning (AL) addresses this bottleneck by strategically selecting the most informative nodes for labeling, typically using uncertainty-based acquisition strategies such as Least Confidence (LC), Entropy, or Bayesian Active Learning by Disagreement (BALD). However, graph-structured data presents unique challenges that distinguish it from traditional active learning settings (Hu et al., 2020). The structural dependencies and non-i.i.d. nature of graphs make uncertainty estimation particularly difficult, as node predictions depend on multi-hop neighborhoods rather than being independent samples (Fuchsgruber et al., 2024; Wang et al., 2024). Distinguishing epistemic uncertainty from aleatoric uncertainty becomes further complicated by neighborhood aggregation effects, often leading to biased node selection and suboptimal labeling strategies (Fuchsgruber et al., 2024).

Test-time augmentation (TTA) has proven highly effective in computer vision for improving uncertainty estimation by generating multiple perturbed views of inputs during inference and aggregating their predic-

tions. While TTA has demonstrated clear benefits for image classification (Gaillochet et al., 2022; Wang et al., 2018; Conde et al., 2023), its application to graph-structured data remains largely unexplored (Ju et al., 2023; Bo et al., 2021). This gap presents a significant opportunity: graph-specific augmentations could exploit relational structure to yield more reliable uncertainty estimates for active learning, potentially addressing the fundamental challenges of uncertainty quantification in graph settings.

We introduce GATTA (Graph Active Learning with Test-Time Augmentation), a framework that systematically integrates test-time augmentation into graph active learning pipelines. GATTA combines graph-specific augmentations with two aggregation strategies, GATTA-S (Score Aggregation) and GATTA-P (Prediction Aggregation), and incorporates a consistency-based filtering mechanism that preserves label-relevant properties while discarding potentially misleading perturbations. Through comprehensive evaluation across multiple datasets, GNN architectures, and AL acquisition strategies, we demonstrate that TTA particularly benefits uncertainty-based methods, enabling simple strategies to achieve competitive performance with complex approaches while reducing computational overhead. Our findings suggest a practical design principle: rather than engineering sophisticated acquisition functions, practitioners can augment efficient uncertainty-based methods with TTA to achieve strong active learning performance at lower computational cost.

Contributions

1. We introduce GATTA, a framework for test-time augmentation in graph active learning with consistency-based filtering for non-label-invariant augmentations.
2. We demonstrate through comprehensive experiments that simple uncertainty methods with GATTA match complex acquisition functions at lower computational cost.
3. We provide practical deployment guidelines covering augmentation type, strength, and ensemble size.

2 Related Work

2.1 Active Learning on Graphs

Active learning on graphs presents fundamental challenges that distinguish it from classical active learning paradigms. The structural interdependencies inherent in graph data violate the independence assumption underlying traditional uncertainty sampling methods, as node predictions are influenced by their multi-hop neighborhoods rather than being isolated samples (Kipf & Welling, 2016). This interconnected nature complicates uncertainty estimation, where distinguishing epistemic uncertainty (reducible through additional labels) from aleatoric uncertainty (irreducible data noise) becomes particularly challenging due to information propagation effects (Wang et al., 2024).

Early graph-specific active learning approaches focused on adapting classical strategies to structural settings. Cai et al. (2017) introduced Active Graph Embedding (AGE), which combined graph embeddings with uncertainty and centrality measures to identify informative nodes. Gao et al. (2018) formulated node selection as a multi-armed bandit problem in their ANRMAB framework, while Regol et al. (2020) proposed Graph Expected Error Minimization (GEEM), directly targeting nodes that maximize expected error reduction across the graph structure.

Recent advances have pursued more sophisticated uncertainty quantification strategies. Kang et al. (2022) proposed JuryGCN, a frequentist-based approach that quantifies uncertainty in GCNs using jackknife estimators. Most relevantly, Fuchsgruber et al. (2024) demonstrated that epistemic uncertainty sampling is theoretically optimal for graph active learning, developing practical approximation methods (Multiple Pseudo-Labels (MP) and Expected Single Pseudo-Label (ESP)) that achieve strong empirical performance.

Despite their promise, these sophisticated methods remain computationally demanding, which limits their scalability and ease of adoption. This raises the question of whether simpler, lightweight strategies, augmented with techniques such as test-time augmentation, can achieve similar or even superior performance at a fraction of the cost.

2.2 Uncertainty Quantification in Graph Neural Networks

Robust uncertainty quantification in GNNs requires addressing the unique challenges posed by relational data structures. Bayesian approaches have shown promise, with Zhang et al. (2018) developing variational inference methods for GNNs and Lakshminarayanan et al. (2016) proposing ensemble-based uncertainty estimation that accounts for graph structure.

Monte Carlo Dropout has been adapted for graph settings by Gal & Ghahramani (2015), who demonstrated improved calibration on node classification tasks. Meanwhile, Zhuang et al. (2024) explored temperature scaling specifically designed for graph neural networks.

These approaches, although effective, often require architectural modifications or additional training procedures, limiting their applicability to existing models.

2.3 Test-Time Augmentation

Test-time augmentation enhances model predictions by generating multiple transformed views of the input during inference and aggregating their outputs. For a model f with parameters θ and input x, TTA computes predictions as:

$$\hat{Y} = \frac{1}{N} \sum_{i=1}^{N} f_\theta(\varphi_i(x)) \tag{1}$$

where φ_i represents different augmentation functions. This ensemble approach provides richer uncertainty estimates by capturing prediction variance across multiple input perturbations, particularly valuable for distinguishing epistemic from aleatoric uncertainty (Wang et al., 2018).

TTA has demonstrated consistent improvements across diverse domains. In computer vision, Shanmugam et al. (2021) showed significant gains in medical image segmentation, while Conde et al. (2023) established connections between TTA and model calibration theory. Natural language processing applications have emerged more recently, with Lu et al. (2022) applying TTA to text classification with word- and character-based augmentations.

Recent work explores TTA specifically for active learning contexts. Gaillochet et al. (2022) demonstrated improved uncertainty estimation for medical image annotation, establishing TTA's potential for query strategy enhancement.

2.4 Graph Augmentation Techniques

Graph augmentation strategies form the foundation for effective TTA in graph settings. Structural augmentations include edge dropout Rong et al. (2019), where edges are randomly removed during training or inference, and graph subsampling that aims to find augmented graph instances from the input graphs that best preserve desired properties by keeping a portion of nodes and their underlying linkages (Qiu et al., 2020). Node-level augmentations, such as feature masking, Gaussian noise injection, and feature shuffling, target different aspects of graph structure preservation (Ding et al., 2022).

The critical challenge in graph augmentation lies in maintaining label-relevant properties. Unlike image transformations such as rotation or cropping, which typically preserve object identity, graph modifications can fundamentally alter node predictions. Yue et al. (2022) addressed this by perturbing graphs in representation space under label-preserving constraints, while Luo et al. (2022) used a reinforcement-learning framework to search for label-invariant policies.

2.5 Novelty and Positioning

While graph augmentation techniques (Liu et al., 2021; Zhao et al., 2020) have been explored for computational efficiency (Cui et al., 2022), representation learning during training (Katsimpras & Paliouras, 2024), and class-balancing via reinforcement learning (Yu et al., 2024), GATTA fundamentally differs in both objective and design. Prior work applies augmentation either as a preprocessing step to scale training (Cui

et al., 2022), to improve self-training performance (Katsimpras & Paliouras, 2024), or to address class imbalance (Yu et al., 2024). Test-time augmentation (TTA) itself has been used in graph learning but never for uncertainty-driven active learning. Bo et al. (2021) applied TTA for social influence prediction and Ju et al. (2023) used virtual node augmentation to address degree bias. GATTA is the first framework to systematically leverage TTA at inference time to generate perturbed graph views, explicitly improving uncertainty quantification across acquisition strategies. This distinction is critical: GATTA is strategy-agnostic and directly enhances uncertainty estimates, whereas existing methods focus on training dynamics or specific acquisition heuristics.

3 Method

3.1 Acquisition Strategies

In active learning, an *acquisition function* (or query strategy) assigns each unlabeled node a score, indicating its informativeness for model improvement. Given predicted class probabilities $P \in \mathbb{R}^{|V| \times C}$, an acquisition function is a mapping $Q : \mathbb{R}^{|V| \times C} \to \mathbb{R}^{|V|}$, where $Q(P)_v$ denotes the informativeness score for node $v \in V$.

We categorize acquisition strategies into three groups: **simple uncertainty-based methods** (Least Confidence (Wang & Shang, 2014), Entropy (Shannon, 1948)), **complex uncertainty-based methods** (MP, ESP (Fuchsgruber et al., 2024)), and **other methods** (AGE (Cai et al., 2017), ANRMAB (Gao et al., 2018), GEEM (Regol et al., 2020)).

3.2 Graph Active Learning with Test-Time Augmentation

We designed GATTA as a plug-and-play module to enhance existing graph active learning strategies without requiring architectural changes. GATTA improves uncertainty estimation by generating multiple perturbed views of the input graph via augmentation and analyzing prediction consistency across these views.

Formally, given an input graph $G = (A, X)$ with adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$ and node features $X \in \mathbb{R}^{|V| \times D}$, we sample N augmented views using a stochastic augmentation function φ:

$$G^{(i)} = \varphi_i(G) = (A^{(i)}, X^{(i)}), \quad i = 1, ..., N \tag{2}$$

along with $G^{(0)} = G$ that denotes the original (unaugmented) graph. Each augmented view $G^{(i)}$ is processed by a pre-trained GNN f_θ to obtain class probability predictions:

$$\mathbf{P}_i = f_\theta(G^{(i)}) \in \mathbb{R}^{|V| \times C}, \tag{3}$$

where $|V|$ is the number of nodes, D is the feature dimension, and C is the number of classes.

The key challenge is how to aggregate these predictions before node selection, as different aggregation strategies can bias active learning toward different node selections. To address this, we explore two integration approaches that we term GATTA-P and GATTA-S. We describe these aggregation strategies in the following subsections.

3.3 Aggregation Methods: GATTA-S and GATTA-P

We propose two complementary strategies for integrating augmented graph views into active learning, differing in when acquisition scores are computed relative to prediction aggregation:

GATTA-P (Prediction Aggregation) first averages predictions across all views, then applies Q once to the aggregated probabilities:

$$\mathbf{Q}_{\mathrm{P}} = Q\left(\mathbb{E}_{G^{(i)} \in \mathcal{G}}[\mathbf{P}_i]\right) \approx Q\left(\frac{1}{N+1} \sum_{i=0}^{N} \mathbf{P}_i\right)$$

GATTA-S (Score Aggregation) applies the acquisition function $Q : \mathbb{R}^{|V|\times C} \to \mathbb{R}^{|V|}$ independently to each $G^{(i)}$ view, then averages the resulting scores:

$$\mathbf{Q}_{\mathrm{S}} = \mathbb{E}_{G^{(i)}\in\mathcal{G}}\big[Q(\mathbf{P}_i)\big] \approx \frac{1}{N+1}\sum_{i=0}^{N} Q(\mathbf{P}_i) \in \mathbb{R}^{|V|}$$

The key distinction is that GATTA-S preserves per-view uncertainty at the expense of higher computational cost, whereas GATTA-P offers greater efficiency but risks losing detailed uncertainty patterns.

3.4 Graph Augmentations

We employ three standard graph augmentation strategies: feature masking (You et al., 2020), feature noising (Zhang et al., 2022), and edge dropout (Rong et al., 2019). Feature masking randomly sets node features to zero with probability p_{mask}, testing model reliance on specific features. Feature noising adds Gaussian noise $\mathcal{N}(0, \sigma^2_{\text{noise}})$ to node features, introducing controlled uncertainty while preserving feature magnitudes. Edge dropout randomly removes edges with probability p_{drop}, testing structural dependencies. These complementary augmentations expose different sources of model uncertainty while aiming to preserve label-relevant information.

3.5 Consistency Filtering

However, the mentioned augmentations can alter node predictions due to their changes in input features or graph structure. Such label variance degrades uncertainty estimation by introducing spurious confidence or false uncertainty, ultimately undermining active learning performance. To address this, we propose a consistency-filtering mechanism that discards augmented views whose predictions conflict with the original graph $G^{(0)}$. Assuming that the pre-trained model's predictions $f_\theta(G^{(0)})$ are reliable, this ensures that only label-preserving perturbations contribute to uncertainty estimation.

For a node v and view i, let $\hat{y}_i^{(v)} = \arg\max_c(\mathbf{P}_i)_{v,c}$ denote its predicted class c, and define:

$$m_i^{(v)} = \mathbb{1}\{\hat{y}_i^{(v)} = \hat{y}_0^{(v)}\}, \tag{4}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function, $\hat{y}_i^{(v)}$ is the predicted class for node v in the i^{th} view, and $m_i^{(v)}$ is a binary consistency mask. And finally, let $\mathbf{m}_i \in \{0,1\}^{|V|}$ contain all node-level filtering weights.

Thus, the filtered acquisition scores for both GATTA-S and GATTA-P are then computed as:

$$\frac{1}{\mathbf{z}}\sum_{i=0}^{N} \mathbf{m}_i \odot Q\left(\mathbf{P}_i\right) \quad \text{and} \quad Q\left(\frac{1}{\mathbf{z}}\sum_{i=0}^{N} (\mathbf{m}_i \mathbf{1}^T) \odot \mathbf{P}_i\right),$$

where $\odot$ denotes element-wise multiplication. Element-wise division by $\mathbf{z} = \sum_{i=0}^{N} \mathbf{m}_i$ serves as a normalization factor that counts the number of consistent views for each node. $\mathbf{1} \in \{1\}^C$ is used to broadcast the mask $\mathbf{m}_i$ across the class dimension. See Appendix B for confidence weighted filtering variant.

4 Sensitivity and Configuration Analysis

While hyperparameter optimization is typically constrained in active learning settings due to the limited availability of labeled data, understanding GATTA's sensitivity to different configurations is crucial for practical deployment. We therefore conducted a systematic study of augmentation types, filtering mechanisms, strength parameters, ensemble size, and runtime. Experiments in this section utilize two representative datasets (CoraML Getoor et al. (2005) and PubMed Namata et al. (2012)), two models (GCN Kipf & Welling (2016) and SGC Wu et al. (2019a)), and uncertainty-based strategies (Entropy and Least Confidence). We present our findings denoted with **F**.

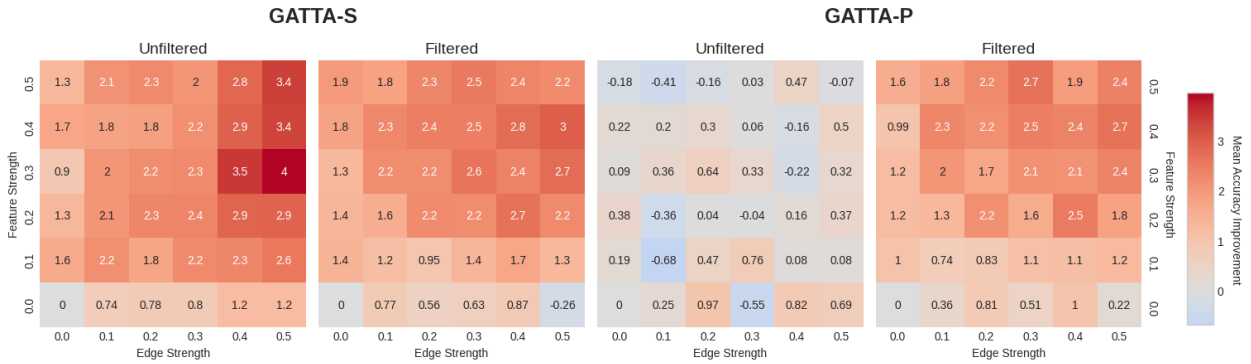

Figure 1: Effect of augmentation strength on performance for FN+ED. Heatmaps show accuracy gains (%) across noise variance ($\sigma^2_{noise} \in [0, 0.5]$) and dropout probability ($p_{drop} \in [0, 0.5]$). Results are reported for GATTA-S (left) and GATTA-P (right), each with and without filtering. Improvements concentrate at higher strengths ($\sim 0.3 - 0.5$). GATTA-S benefits without filtering but filtering broadens the region of effective strengths, while GATTA-P requires filtering to achieve any improvement.

Table 1: Comparison of augmentation types and combinations. Reported are the average and 75 percentile performance gains (%) across datasets, models, and augmentation strengths. Combined augmentations, particularly FN+ED, consistently outperform single augmentations.

Strategy	Average	75th percentile
Feature Masking	0.03 ± 0.83	0.62
Feature Noising	$\mathbf{0.69 \pm 0.67}$	**1.21**
Edge Dropout	0.53 ± 0.84	1.03
FM + ED	0.53 ± 0.96	1.16
FN + ED	$\mathbf{1.12 \pm 1.05}$	**1.94**

4.1 Augmentation Type Selection

We evaluated Feature Masking (FM), Feature Noising (FN), Edge Dropout (ED), and their pairwise combinations across the mentioned models, datasets, and strategies. Table 1 summarizes averaged, augmentation-wise performance.

F1. Feature Noising outperforms Feature Masking ($0.69 \pm 0.67\%$ vs. $0.03 \pm 0.83\%$). Additive noise preserves feature scale during neighborhood aggregation, whereas masking creates information voids that propagate through message-passing layers, making FN more suitable for test-time perturbations.

F2. Combined augmentations consistently outperform single augmentations. The FN+ED combination achieved the largest average improvement ($+1.12 \pm 1.05\%$). FM+ED showed modest gains ($+0.53 \pm 0.96\%$), comparable to single augmentations. Multi-modal perturbations produce more informative uncertainty signals when both augmentation types contribute meaningfully to the ensemble.

Based on these findings, we adopt FN+ED for all subsequent experiments.

4.2 Augmentation Strength and Filtering

We systematically evaluated Feature Noising variance $\sigma^2_{noise} \in [0.0, 0.5]$ and Edge Dropout probability $p_{drop} \in [0.0, 0.5]$, testing all combinations with and without filtering for both GATTA variants. Figure 1 visualizes the interaction between augmentation strength, filtering, and aggregation strategy.

F3. Optimal performance occurs at higher augmentation strengths. With filtering applied, performance peaked at $\sigma^2_{noise} \in [0.3, 0.5]$ and $p_{drop} \in [0.3, 0.5]$, achieving improvements up to 3.0 over baseline

(Figure 1). Stronger perturbations expose more informative uncertainty signals, provided label-inconsistent augmentations are filtered.for For more details, see Appendix C. For comparison with confidence weighted filtering variant, see Appendix B.

F4. GATTA-S consistently outperforms GATTA-P, resulting in different filtering requirements. GATTA-P exhibits strong sensitivity to filtering: without it, the method shows negligible improvement at weak augmentation strengths and active degradation at strong strengths (Figure 1, up to -0.5% decline), failing overall ($0.12 \pm 1.04\%$) but achieving substantial improvements with filtering ($1.96 \pm 1.27\%$). Prediction-level aggregation directly averages class probabilities across views, so label-inconsistent augmentations corrupt the uncertainty estimate, making filtering essential. In contrast, GATTA-S demonstrates robustness across the entire configuration space, performing best without filtering ($2.49 \pm 1.42\%$) and maintaining stable performance even at strong augmentation levels. Score-level aggregation computes acquisition scores independently per view before averaging, allowing the ensemble to balance label-inconsistent signals without explicit filtering. Across all configurations, GATTA-S achieves greater improvements both on average and at peak performance.

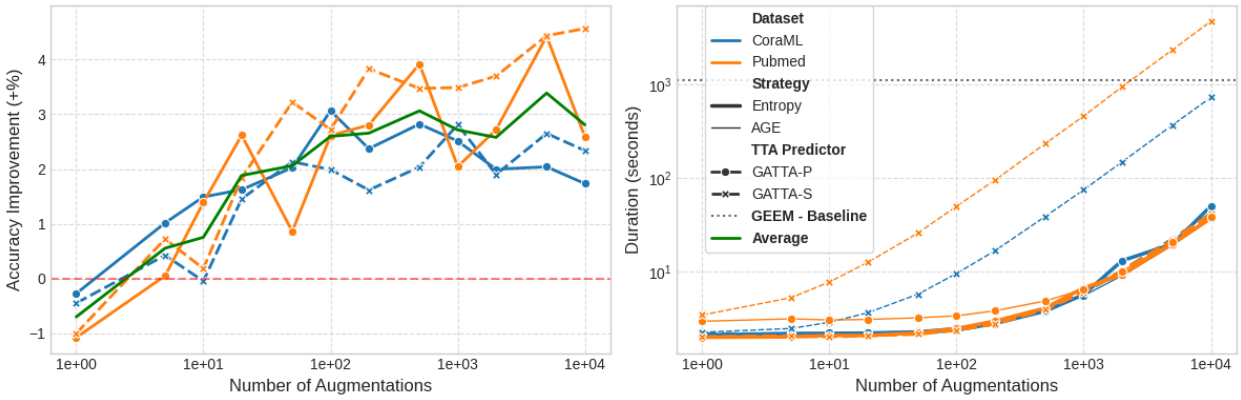

Figure 2: Accuracy improvements (**Left**) and runtime scaling (**Right**) with increasing ensemble size. Accuracy gains saturate around $N \approx 200$. GATTA-S runtime increases drastically for complex strategies (AGE), while both GATTA variants scale well for simple strategies (Entropy)

4.3 Ensemble Size, Runtime, and Scalability Analysis

We evaluated ensemble sizes $N \in [1, 10\,000]$ to characterize the performance-cost trade-off. Figure 2 shows accuracy and runtime scaling patterns.

F5. Performance scales logarithmically with ensemble size. Accuracy improvements follow Performance $\approx$ baseline$+0.37\times\log(N+1)$ (Figure 2, right). Substantial gains occur up to $N \approx 200$ ($+2.65\pm0.93\%$), with diminishing returns thereafter. Beyond $N = 500$, each additional 100 views yields less than 0.1% improvement, suggesting that moderate ensemble sizes capture most of the uncertainty signal diversity.

F6. GATTA runtime scales efficiently. For simple acquisition functions like Entropy, where inference dominates, GATTA-P and GATTA-S scale similarly, both requiring $\sim 2\times$ baseline time at N = 500 (Figure 2, right). For expensive acquisition functions like AGE, where acquisition computation dominates, GATTA-S scales poorly (requiring N+1 AGE evaluations), whereas GATTA-P remains efficient. Crucially, runtime scaling remains consistent across CoraML (2,810 nodes, 15,962 edges) and PubMed (19,717 nodes, 88,648 edges), indicating that GATTA's computational overhead is governed by ensemble size and acquisition complexity rather than graph scale. For a detailed discussion about computational complexity, see Appendix E.

Table 2: Detailed performance evaluation of GATTA configurations. The table presents average accuracy improvements for GATTA-S (Entropy-S, LC-S) and GATTA-P across diverse datasets, GNN models, and acquisition strategies, highlighting key performance patterns. Values represent average accuracy improvement over baseline methods. MP, ESP, and GEEM were only tested with SGC, as in their original implementation. Additionally, ESP and GEEM were not evaluated on the AmazonComputers dataset due to computational limitations.

Dataset	Simple								Complex		Other				
	Entropy-P		Entropy-S		LC-P		LC-S		MP	ESP	ANRMAB		AGE		GEEM
	GCN	SGC	GCN	SGC	GCN	SGC	GCN	SGC	SGC	SGC	GCN	SGC	GCN	SGC	SGC
Citeseer	2.03	3.26	1.82	4.47	0.68	2.04	1.55	3.40	−1.88	−0.53	0.17	0.05	−0.14	0.16	−1.35
CoraML	2.89	1.75	1.57	4.12	2.56	1.02	3.14	3.30	3.96	0.28	2.09	1.53	−0.89	0.05	−1.77
PubMed	4.80	0.52	4.58	2.58	5.09	1.06	6.66	3.17	3.91	−0.54	2.71	0.79	0.81	0.16	−0.22
Amazon Photos	−1.06	1.23	2.20	5.60	2.47	0.54	3.98	3.76	1.95	0.03	0.87	0.68	0.12	0.61	−0.16
Amazon Computers	−0.02	0.90	4.52	5.39	0.84	3.14	3.21	5.77	2.61	–	−1.53	−0.70	−0.09	0.10	–

5 Results

This section presents the empirical evaluation of GATTA across five graph datasets, two GNN architectures, and six active learning acquisition strategies. We evaluate citation networks (Citeseer, CoraML, PubMed) and co-purchase networks (AmazonPhotos, AmazonComputers) using GCN and SGC architectures. Acquisition strategies are grouped into simple uncertainty-based (Least Confidence, Entropy), complex uncertainty-based (MP, ESP), and other methods (AGE, ANRMAB), with GEEM serving as a complex baseline. All results represent averages over 25 independent trials (5 initial pools × 5 initializations). We create 500 augmented views with Feature Noising ($\sigma^2_{\text{noise}} = 0.4$) and Edge Dropout ($p_{\text{drop}} = 0.5$). Based on our sensitivity analysis, we apply filtering only for GATTA-P.

Table 2 summarizes accuracy improvements over baseline, while Figure 3 illustrates performance trajectories on CoraML. See more about the active learning protocol, datasets, models, and training details in Appendix A.

R1. GATTA selectively benefits uncertainty-based methods. Simple uncertainty methods achieve substantial improvements: Least Confidence gains +2.87% average and Entropy gains +3.03%, with GATTA-S peaks exceeding +5% on multiple datasets. Figure 3 shows GATTA-enhanced Entropy and LC exceeding baseline GEEM performance. GATTA-S consistently outperforms GATTA-P for these methods, achieving approximately double the improvement by preserving per-view uncertainty through score-level aggregation. Complex methods show inconsistent results: MP achieves +2.11% average with strong gains on CoraML, PubMed, and AmazonComputers, but declines significantly on Citeseer (-1.88%). ESP averages -0.19%, declining on two datasets, suggesting potential interference between epistemic uncertainty estimation and test-time augmentation. Non-uncertainty methods derive minimal benefit: ANRMAB achieves +0.67% average while AGE shows near-zero improvement (+0.09%) and GEEM actively declines (-0.88%). Overall, methods with sophisticated uncertainty mechanisms or non-uncertainty-based selection derive inconsistent benefit from test-time augmentation.

R2. GATTA's benefits emerge early and persist throughout learning. Figure 3 shows that GATTA improves sample efficiency from the earliest acquisition rounds, when model uncertainty is highest and label budgets most constrained. The performance gap between GATTA-enhanced and baseline methods remains consistent across the learning trajectory, indicating that GATTA does not merely accelerate early learning but provides sustained improvement. This is particularly valuable in practical settings where labeling budgets are exhausted before model saturation. For a more detailed discussion about learning dynamics and performance, see Appendix D.

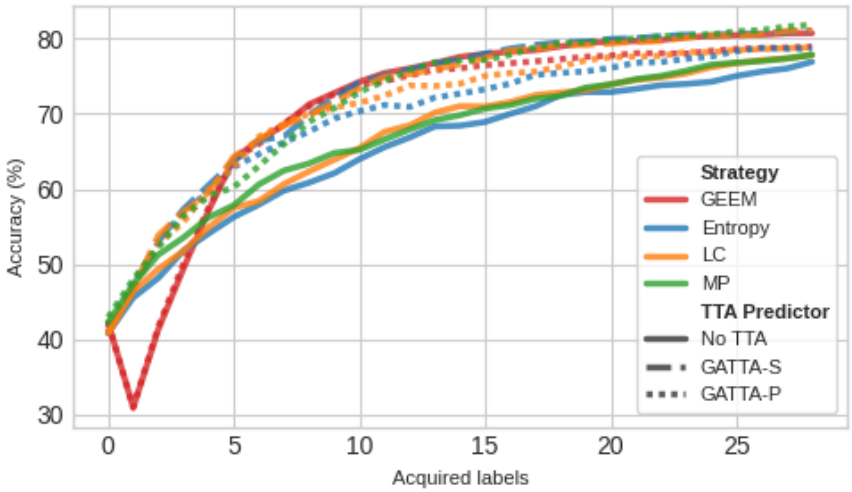

Figure 3: Learning curves across acquisition strategies on the CoraML dataset. TTA (dashed: GATTA-P, dotted: GATTA-S) improves sample efficiency for uncertainty-based strategies such as Entropy and LC, while having a limited effect on structure-based strategies like GEEM.

R3. Performance depends on unique dataset characteristics, not broader graph category. While citation and co-purchase networks achieve comparable average improvements, this aggregate masks substantial within-category variance. Among citation networks, performance ranges from strong gains on PubMed to modest improvements on Citeseer. GATTA-S with simple uncertainty methods achieves exceptional performance on co-purchase networks, with improvements exceeding +5% for both Entropy-S and LC-S on AmazonComputers using SGC. This variation suggests that specific graph properties, such as homophily, feature informativeness, or class structure, influence GATTA's effectiveness more than broad domain categories. We recommend pilot testing on a data subset before full deployment.

R4. GATTA generalizes across architectures. Our main experiments evaluate GATTA across six acquisition strategies using GCN and SGC. To test architectural generalizability, we additionally evaluate on GAT and GraphSAGE with Entropy acquisition. GATTA generalizes effectively across all four architectures, with both GAT and GraphSAGE showing consistent improvements on most datasets (Table 3, right). GATTA-S remains the stronger variant for GAT, while GraphSAGE exhibits more dataset-dependent behavior between GATTA-P and GATTA-S. These results confirm that GATTA functions as an architecture-agnostic module that operates at the input level, requiring no model modifications.

R5. GATTA outperforms MC-Dropout MC Dropout is one of the most widely used methods for uncertainty estimation in neural networks without requiring ensemble training, making it a natural baseline for comparison. GATTA matches or exceeds MC Dropout performance on 4 of 5 datasets, with gains up to +3.73% on PubMed (Table 3, left). However, combining GATTA with MCD yields no consistent improvement and can substantially degrade performance (GATTA-P + MCD drops performance on AmazonComputers by 8.39% below baseline). This suggests that both methods capture overlapping uncertainty information, and their combination introduces redundant or conflicting signals. Since GATTA operates at the input level without requiring architectural modifications, it offers a simpler and more effective alternative to dropout-based uncertainty estimation for graph active learning.

6 Discussion and Conclusions

This work introduces GATTA, a framework for systematically integrating test-time augmentation into graph active learning. Our evaluation reveals a practical design principle: practitioners can achieve competitive performance by augmenting simple uncertainty methods with test-time augmentation, rather than engineering sophisticated acquisition functions. Simple methods enhanced with GATTA achieve +2.9–3.0% average

Table 3: (**Left**) Accuracy gains over Entropy baseline with MC Dropout (MCD) on GCN model. GATTA-P and GATTA-S consistently match or outperform MCD, while their combination yields no additive benefit. (**Right**) GATTA performance gains across GNN architectures, confirming architecture-agnostic design. Best per dataset/architecture in bold.

GATTA	MCD	Am. Co.	Am. Ph.	Cite.	Cora	PubM.
–	✓	**+5.71**	-0.49	+1.32	+1.93	+1.07
P	–	-0.02	+0.33	+1.13	**+2.89**	**+4.80**
	✓	-8.39	-0.98	**+2.03**	+0.45	+2.00
S	–	-1.52	**+2.20**	+1.87	+2.09	+3.70
	✓	+4.24	-0.30	+1.78	+1.36	+2.85

Model	GATTA	Am. Co.	Am. Ph.	Cite.	Cora	PubM.
GCN	P	-0.02	-1.06	+2.03	+2.89	+4.80
	S	+4.52	+2.20	+1.82	+1.58	+4.58
SGC	P	+0.90	+1.23	+3.26	+1.75	+0.52
	S	+5.39	+5.60	+4.47	+4.12	+2.58
GAT	P	+3.24	+3.05	+2.83	+0.74	+3.81
	S	+5.39	+3.32	+3.05	+2.56	+1.71
SAG	P	+2.18	-0.19	-0.35	-2.61	-0.72
	S	+1.93	+1.56	+1.66	+0.41	-1.21

improvement, with peaks exceeding +5%, matching or exceeding complex baselines while requiring no architectural modifications. GATTA succeeds by exposing model sensitivity to local perturbations: nodes where predictions remain stable across augmented views are likely well-understood, while inconsistent predictions reveal uncertainty worthy of labeling. This input-level approach complements existing model-level uncertainty methods and offers a flexible, architecture-agnostic alternative.

For practitioners, we offer four guidelines. First, prioritize simple uncertainty methods (Entropy, Least Confidence), as they benefit most from TTA and achieve performance competitive with complex baselines at substantially lower implementation effort. Second, use GATTA-S whenever computationally feasible, as it consistently outperforms GATTA-P and requires no filtering; resort to GATTA-P with filtering only when acquisition functions are expensive. Third, use moderate-to-high augmentation strength ($\sigma^2_{\text{noise}} \in [0.4, 0.5]$, $p_{\text{drop}} \in [0.3, 0.5]$) with ensemble size $N = 500$, which captures most uncertainty signal diversity while maintaining practical runtime. Fourth, expect dataset-dependent results, so pilot testing on a data subset is advisable.

Several limitations bound our findings. Our evaluation focuses exclusively on node classification in transductive settings; extensions to link prediction, graph classification, or inductive scenarios remain unexplored. Our empirical study spans five datasets from homophilic citation and co-purchase networks. We expect GATTA's principles to generalize, as the core mechanism, measuring prediction stability under input perturbations, does not assume specific graph structure. Other graph types, such as dynamic or heterophilic graphs, present distinct challenges that warrant dedicated investigation beyond our current scope. Our filtering mechanism assumes reliable base model predictions, which may not hold in early active learning rounds. We provide no theoretical guarantees on GATTA's convergence or sample complexity.

Future work should investigate adaptive augmentation strategies that dynamically adjust perturbation strength based on model confidence or graph structure. Theoretical analysis connecting GATTA's effectiveness to graph properties such as homophily or feature informativeness would provide principled guidance for deployment. The negative interaction between TTA and epistemic uncertainty methods like ESP warrants further investigation.

Beyond active learning, our findings suggest that test-time augmentation may be broadly applicable for uncertainty quantification in graph neural networks. GATTA demonstrates that test-time augmentation transforms simple acquisition functions into competitive alternatives to complex methods. By making graph active learning more accessible to practitioners while establishing TTA as a broadly applicable tool, this work opens avenues for test-time augmentation across graph machine learning.

References

Hongbo Bo, Ryan McConville, Jun Hong, and Weiru Liu. Social Influence Prediction with Train and Test Time Augmentation for Graph Neural Networks. In *2021 International Joint Conference on Neural*

Networks (IJCNN), pp. 1–8, 2021. doi: 10.1109/IJCNN52387.2021.9533437. URL https://ieeexplore.ieee.org/document/9533437/?arnumber=9533437.

Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. Active Learning for Graph Embedding. *arXiv preprint arXiv:1705.05085*, 2017. doi: 10.48550/arXiv.1705.05085. URL http://arxiv.org/abs/1705.05085.

Pedro Conde, Tiago Barros, Rui L. Lopes, Cristiano Premebida, and Urbano J. Nunes. Approaching Test Time Augmentation in the Context of Uncertainty Calibration for Deep Neural Networks. *arXiv preprint arXiv:2304.05104*, 2023. doi: 10.48550/arXiv.2304.05104. URL http://arxiv.org/abs/2304.05104.

Limeng Cui, Xianfeng Tang, Sumeet Katariya, Nikhil Rao, Pallav Agrawal, Karthik Subbian, and Dongwon Lee. Allie: Active learning on large-scale imbalanced graphs. In *Proceedings of the ACM web conference 2022*, pp. 690–698, 2022.

Kaize Ding, Zhe Xu, Hanghang Tong, and Huan Liu. Data Augmentation for Deep Graph Learning: A Survey. *ACM SIGKDD Explorations Newsletter*, 2022. doi: 10.48550/arXiv.2202.08235. URL http://arxiv.org/abs/2202.08235.

Dominik Fuchsgruber, Tom Wollschläger, Bertrand Charpentier, Antonio Oroz, and Stephan Günnemann. Uncertainty for Active Learning on Graphs. In *International Conference on Machine Learning*, 2024. doi: 10.48550/arXiv.2405.01462. URL http://arxiv.org/abs/2405.01462.

Mélanie Gaillochet, Christian Desrosiers, and Hervé Lombaert. TAAL: Test-time Augmentation for Active Learning in Medical Image Segmentation. In *Data Augmentation, Labelling, and Imperfections (DALI@MICCAI)*, pp. 43–53, 2022. doi: 10.1007/978-3-031-17027-0_5.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *International Conference on Machine Learning*, 2015. doi: 10.48550/arXiv.1506.02142. URL http://arxiv.org/abs/1506.02142.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian Active Learning with Image Data. In *International Conference on Machine Learning*, 2017. doi: 10.48550/arXiv.1703.02910. URL http://arxiv.org/abs/1703.02910.

Li Gao, Hong Yang, Chuan Zhou, Jia Wu, Shirui Pan, and Yue Hu. Active Discriminative Network Representation Learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2142–2148, 2018. doi: 10.24963/ijcai.2018/296. URL https://www.ijcai.org/proceedings/2018/296.

Lise Getoor, Sanghamitra Bandyopadhyay, Ujjwal Maulik, Lawrence Holder, and Diane Cook. Link-based classification. In *Advanced Methods for Knowledge Discovery from Complex Data*, pp. 189–207. Springer-Verlag, 01 2005. ISBN 1-85233-989-6. doi: 10.1007/1-84628-284-5_7.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*, 2017. doi: 10.48550/arXiv.1704.01212. URL http://arxiv.org/abs/1704.01212.

Asmaa Halbouni, Teddy Surya Gunawan, Mohamed Hadi Habaebi, Murad Halbouni, Mira Kartiwi, and Robiah Ahmad. Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. *IEEE Access*, 10:19572–19585, 2022. ISSN 2169-3536. doi: 10.1109/ACCESS.2022.3151248. URL https://ieeexplore.ieee.org/document/9712274.

Shengding Hu, Zheng Xiong, Meng Qu, Xingdi Yuan, Marc-Alexandre Côté, Zhiyuan Liu, and Jian Tang. Graph Policy Network for Transferable Active Learning on Graphs. In *Neural Information Processing Systems*, 2020. doi: 10.48550/arXiv.2006.13463. URL http://arxiv.org/abs/2006.13463.

Mingxuan Ju, Tong Zhao, Wenhao Yu, Neil Shah, and Yanfang Ye. GraphPatcher: Mitigating Degree Bias for Graph Neural Networks via Test-time Augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. doi: 10.48550/arXiv.2310.00800. URL http://arxiv.org/abs/2310.00800.

Jian Kang, Qinghai Zhou, and Hanghang Tong. JuryGCN: Quantifying Jackknife Uncertainty on Graph Convolutional Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 742–752, 2022. doi: 10.1145/3534678.3539286. URL http://arxiv.org/abs/2210.05959.

Georgios Katsimpras and Georgios Paliouras. Improving graph neural networks by combining active learning with self-training. *Data Mining and Knowledge Discovery*, 38(1):110–127, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2014. doi: 10.48550/arXiv.1412.6980. URL http://arxiv.org/abs/1412.6980.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2016. doi: 10.48550/arXiv.1609.02907. URL http://arxiv.org/abs/1609.02907.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. doi: 10.48550/arXiv.1612.01474. URL http://arxiv.org/abs/1612.01474.

Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

Songtao Liu, Rex Ying, Hanze Dong, Lanqing Li, Tingyang Xu, Yu Rong, Peilin Zhao, Junzhou Huang, and Dinghao Wu. Local Augmentation for Graph Neural Networks. In *International Conference on Machine Learning*, 2021. doi: 10.48550/arXiv.2109.03856. URL http://arxiv.org/abs/2109.03856.

Helen Lu, Divya Shanmugam, Harini Suresh, and John Guttag. Improved Text Classification via Test-Time Augmentation. *arXiv preprint arXiv:2206.13607*, 2022. doi: 10.48550/arXiv.2206.13607. URL http://arxiv.org/abs/2206.13607.

Youzhi Luo, Michael McThrow, Wing Yee Au, Tao Komikado, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Automated Data Augmentations for Graph Classification. In *International Conference on Learning Representations (ICLR)*, 2022. doi: 10.48550/arXiv.2202.13248. URL http://arxiv.org/abs/2202.13248.

Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. Query-driven active surveying for collective classification. In *10th international workshop on mining and learning with graphs*, volume 8, pp. 1, 2012.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1150–1160, 2020. doi: 10.1145/3394486.3403168. URL http://arxiv.org/abs/2006.09963.

Florence Regol, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Active Learning on Attributed Graphs via Graph Cognizant Logistic Regression and Preemptive Query Generation. In *International Conference on Machine Learning*, 2020. doi: 10.48550/arXiv.2007.05003. URL http://arxiv.org/abs/2007.05003.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *International Conference on Learning Representations*, 2019. doi: 10.48550/arXiv.1907.10903. URL http://arxiv.org/abs/1907.10903.

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. ISSN 0738-4602, 2371-9621. doi: 10.1609/aimag.v29i3.2157. URL https://onlinelibrary.wiley.com/doi/10.1609/aimag.v29i3.2157.

Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1214–1223, 2021.

C E Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of Graph Neural Network Evaluation. *arXiv preprint arXiv:1811.05868*, 2018. doi: 10.48550/arXiv.1811.05868. URL http://arxiv.org/abs/1811.05868.

Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 112–119. IEEE, 2014. doi: 10.1109/IJCNN.2014.6889457. URL https://ieeexplore.ieee.org/document/6889457.

Fangxin Wang, Yuqing Liu, Kay Liu, Yibo Wang, Sourav Medya, and Philip S. Yu. Uncertainty in Graph Neural Networks: A Survey. *arXiv preprint arXiv:2403.07185*, 2024. doi: 10.48550/arXiv.2403.07185. URL http://arxiv.org/abs/2403.07185.

Guotai Wang, Wenqi Li, Michael Aertsen, Jan Deprest, Sebastien Ourselin, and Tom Vercauteren. Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*, 338:34–45, 2018. ISSN 09252312. doi: 10.1016/j.neucom.2019.01.103. URL http://arxiv.org/abs/1807.07356.

Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q. Weinberger. Simplifying Graph Convolutional Networks. In *International Conference on Machine Learning (ICML)*, 2019a. doi: 10.48550/arXiv.1902.07153. URL http://arxiv.org/abs/1902.07153.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chemical Science*, 2017. doi: 10.48550/arXiv.1703.00564. URL http://arxiv.org/abs/1703.00564.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1): 4–24, 2019b. ISSN 2162-237X, 2162-2388. doi: 10.1109/TNNLS.2020.2978386. URL http://arxiv.org/abs/1901.00596.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph Contrastive Learning with Augmentations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. doi: 10.48550/arXiv.2010.13902. URL http://arxiv.org/abs/2010.13902.

Chengcheng Yu, Jiapeng Zhu, and Xiang Li. Graphcbal: Class-balanced active learning for graph neural networks via reinforcement learning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 3022–3031, 2024.

Han Yue, Chunhui Zhang, Chuxu Zhang, and Hongfu Liu. Label-invariant Augmentation for Semi-Supervised Graph Classification. In *Neural Information Processing Systems*, 2022. URL http://arxiv.org/abs/2205.09802.

Yifei Zhang, Hao Zhu, Zixing Song, Piotr Koniusz, and Irwin King. COSTA: Covariance-Preserving Feature Augmentation for Graph Contrastive Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2524–2534, 2022. doi: 10.1145/3534678.3539425. URL http://arxiv.org/abs/2206.04726.

Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Üstebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *AAAI Conference on Artificial Intelligence*, number arXiv:1811.11103. arXiv, November 2018. doi: 10.48550/arXiv.1811.11103.

Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. Data Augmentation for Graph Neural Networks. In *AAAI Conference on Artificial Intelligence*, 2020. doi: 10.48550/arXiv.2006.06830. URL http://arxiv.org/abs/2006.06830.

Jie Zhou, Ganqu Cui, Shengding Hu, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph Neural Networks: A Review of Methods and Applications. *AI Open*, 1: 57–81, 2020. doi: 10.48550/arXiv.1812.08434. URL http://arxiv.org/abs/1812.08434.

Dingyi Zhuang, Chonghe Jiang, Yunhan Zheng, Shenhao Wang, and Jinhua Zhao. GETS: Ensemble Temperature Scaling for Calibration in Graph Neural Networks. In *International Conference on Learning Representations*, 2024. doi: 10.48550/arXiv.2410.09570. URL http://arxiv.org/abs/2410.09570.

A Experimental Setup

We adopt the codebase and the experimental configuration from Fuchsgruber et al. (2024), including training procedures, datasets, and active learning protocols. Our implementation of GATTA can be found in this repository: https://anonymous.4open.science/r/gatta-8A38. Below, we describe the key components of our setup.

A.1 Active Learning Protocol

We conduct active learning on graphs where, given an initial set of labeled nodes $\mathcal{L} \subset V$, we aim to acquire labels for unlabeled nodes $\mathcal{U} \subset V$ to maximize classifier performance. Our protocol proceeds as follows:

1. A single node is randomly drawn from each class to form the initial training set.
2. The model is initialized and trained until convergence.
3. The acquisition strategy selects an unlabeled node for labeling.
4. We add the acquired label to the training set and repeat from step (2) until the acquisition budget is exhausted.
5. After the final acquisition round, the model is retrained on all labeled nodes, and its classification accuracy is reported on the held-out test set as the final performance metric.

Following Fuchsgruber et al. (2024), we re-train the classifier from scratch after each acquisition iteration. Unless stated otherwise, we acquire one node label per iteration and fix the acquisition budget to $4C$, where C is the number of classes. The resulting final training pools, therefore, contain fewer instances compared to dataset splits commonly used in standard semi-supervised learning benchmarks.

A.2 Datasets

We evaluate our approach on standard node classification benchmark datasets from the literature. Following Fuchsgruber et al. (2024), we consider three citation networks and two co-purchase networks:

Citation Networks: CoraML (Getoor et al., 2005), Citeseer (Sen et al., 2008), and PubMed (Namata et al., 2012). In these datasets, nodes represent papers and edges represent citations.

Co-purchase Networks: AmazonComputers and AmazonPhotos (Shchur et al., 2018). In these datasets, nodes represent products and edges indicate that products are frequently co-purchased.

Dataset statistics are provided in Table 4.

Table 4: Dataset statistics. Homophily measures the fraction of edges connecting nodes of the same class.

Dataset	#Nodes	#Edges	#Features	#Classes	Edge Density	Homophily
CoraML	2,810	15,962	2,879	7	0.20%	78.44%
Citeseer	1,681	5,804	602	6	0.20%	92.76%
PubMed	19,717	88,648	500	3	0.02%	80.24%
AmazonComputers	13,381	491,556	767	10	0.27%	77.72%
AmazonPhotos	7,484	238,086	745	8	42.47%	82.72%

A.3 Model Details

As hyperparameter tuning may be unrealistic in active learning settings (Regol et al., 2020), we do not perform validation-based hyperparameter optimization for our models. Instead, we adopt hyperparameters reported as effective in the literature and apply them uniformly across all datasets. Specifically, we use the configuration for both SGC (Wu et al., 2019a) and GCN (Kipf & Welling, 2016) from Table 5.

Table 5: Hyperparameters for GNN models.

Layers	Hidden Dim.	Learning Rate	Max Epochs	Weight Decay	Dropout
1	[64]	0.001	10,000	0.001	0.8

A.4 Training and Evaluation Details

We train all models using the binary cross-entropy loss with the Adam optimizer (Kingma & Ba, 2014), learning rate of 10^{-3}, and weight decay of 10^{-3}. We perform early stopping on validation loss with patience of 100 iterations.

For each dataset, acquisition function, and model, we evaluate five dataset splits with five independent model initializations each (25 runs in total), and report the averaged results. A priori, we fix 20% of all nodes as a test set that is reused across all splits and initializations and cannot be acquired by any strategy. For each dataset split, we fix 20% of all nodes as a validation set and reuse it across initializations.

We report test accuracy as our primary performance metric. All reported accuracy gains are computed with respect to the corresponding non-TTA baseline, i.e., the same dataset-model-acquisition configuration evaluated without test-time augmentation.

B Filtering variants

The filtering mechanism described in Section 3.5, enforces strict prediction consistency through binary masks, which we refer to as *Hard filtering* from now on. We additionally explored a confidence-weighted variant motivated by the intuition that not all consistent predictions should contribute equally. Instead of binary masks, we compute soft weights based on each view's confidence for the original prediction's class:

$$s_i^{(v)} = (\mathbf{P}_i)_{v,c^\star}, \quad \text{where} \quad c^\star = \arg\max_c (\mathbf{P}_0)_{v,c} \tag{5}$$

Using these soft weights alone proved ineffective, as inconsistent views with spuriously high confidence for the wrong class introduced noise. However, combining hard and soft filtering, discarding inconsistent views while weighting consistent ones by confidence, yielded a viable alternative we term *Firm filtering*:

$$w_i^{(v)} = m_i^{(v)} \cdot s_i^{(v)} \tag{6}$$

Since $m_i^{(v)}$ masks out inconsistent predictions, firm filtering reduces to weighting each view by its confidence for the predicted class. Despite early promising results, firm filtering did not outperform hard filtering in our experiments (Table 6), suggesting that uniform weighting of consistent views is sufficient. We report firm filtering results in Figures 10–13 for completeness.

Table 6: Performance comparison (% accuracy gain over baseline) for GATTA variants under different filtering strategies. Results averaged across CoraML and PubMed with GCN and SGC models using Entropy and LC.

Filtering	GATTA-P	GATTA-S	*Average*
Hard	1.96 ± 1.27	2.16 ± 1.28	2.06 ± 1.28
Firm	1.97 ± 1.34	2.14 ± 1.35	2.06 ± 1.34
None	0.12 ± 1.04	2.49 ± 1.42	1.31 ± 1.72
Average	1.37 ± 1.5	2.26 ± 1.36	1.81 ± 1.50

C Impact of Data Augmentation

In this section, we provide a detailed analysis of augmentation strength experiments for Feature Noising and Edge Drop across two datasets (CoraML, PubMed) and two acquisition strategies (Entropy, LC), as referenced in Section 4.2. Figures 10– 13 report accuracy gains relative to the corresponding non-TTA baseline, that is, the same dataset-model-acquisition configuration evaluated without test-time augmentation.

As discussed in Section 4.2, stronger augmentations generally yield better performance. Notably, filtering mechanisms play distinct roles depending on the aggregation strategy: for GATTA-P, consistency-based filtering is essential to maintain performance, particularly at higher augmentation strengths, whereas GATTA-S achieves optimal results without filtering. These trends are consistent across both SGC and GCN architectures. Additionally, we observe no substantial performance difference between Firm and Hard filtering variants.

D Learning Dynamics and Performance Comparison

Figures 4– 8 present learning curves across acquisition strategies, GNN architectures, and datasets. The upper row displays results for GCN, while the bottom row shows SGC performance. The left column presents simple uncertainty-based strategies (Entropy, LC), and the right column shows complex strategies (AGE, ANRMAB, ESP, MP, GEEM). Following the original implementations (Fuchsgruber et al., 2024; Regol et al., 2020), ESP, MP, and GEEM were evaluated exclusively with SGC. Table 7 summarizes the final test accuracies after all active learning iterations, providing a quantitative complement to the learning curve visualizations.

Both GATTA variants substantially improve simple strategies, enabling them to close the performance gap with GEEM and, in several cases, surpass it (Citeseer, CoraML, PubMed). The improvements are most pronounced for simple uncertainty-based methods and, notably, for the MP strategy. Across architectures, GATTA-P and GATTA-S demonstrate consistent benefits, with GATTA-S often achieving better performance in later iterations, particularly on citation networks.

Dataset characteristics significantly influence GATTA's effectiveness, though not along simple domain boundaries. Within citation networks, performance ranges from strong gains on PubMed to modest improvements on Citeseer. Co-purchase networks show high variance: Amazon Computers achieves exceptional improvements (+5% or more with GATTA-S), while Amazon Photos shows more moderate gains. This variation suggests that specific graph properties—such as feature informativeness, homophily, or class balance—matter more than broad dataset categories.

For complex strategies like AGE and ANRMAB, GATTA provides more modest improvements and, in some cases, may degrade performance. This suggests that sophisticated acquisition functions already incorporate mechanisms that partially account for prediction uncertainty, making additional augmentation-based refinement less beneficial. Similarly, GEEM shows consistent slight degradation with GATTA, indicating potential interference between structure-aware acquisition and input-level perturbations.

Beyond accuracy improvements, GATTA consistently reduces performance variance. Across all simple method configurations (Entropy and LC with GCN and SGC), GATTA-S reduces standard deviation in 18 of 20 dataset-architecture combinations, with the exceptions occurring on PubMed with SGC. This indicates more reliable uncertainty estimation: aggregating predictions across augmented views stabilizes node selection, yielding more consistent outcomes across initializations.

Table 7: Test accuracy (%) across datasets, acquisition strategies, and GNN architectures. Results compare baseline strategies (no GATTA indicator) with GATTA-P and GATTA-S variants using GCN and SGC architectures. Bold indicates best performance per strategy-dataset combination. Standard deviations computed over 25 (5 dataset initializations × 5 model initializations) runs. Missing entries indicate experiments not conducted for that configuration due to computational limitations.

		Citeseer		CoraML		PubMed		Amazon Photos		Amazon Computers	
Strategy	GATTA	GCN	SGC	GCN	SGC	GCN	SGC	GCN	SGC	GCN	SGC
Entropy	–	87.48±3.37	84.67±6.29	74.61±4.61	76.89±4.06	67.69±6.01	67.37±5.79	84.47±5.99	79.79±8.91	70.06±7.81	69.18±6.63
	P	89.51±1.60	87.93±1.81	77.50±2.67	78.64±2.83	72.49±3.53	67.89±7.36	83.41±6.34	81.02±7.98	70.04±6.91	70.08±7.24
	S	89.30±1.35	89.14±1.22	76.19±3.35	81.01±1.63	72.27±5.49	69.95±6.75	86.67±3.83	85.39±5.99	74.58±6.51	74.57±4.20
LC	–	87.83±2.26	86.45±2.61	74.49±4.05	77.79±2.65	66.72±7.27	68.59±4.92	84.20±5.93	81.19±7.66	73.38±5.47	70.68±6.64
	P	88.52±2.11	88.49±1.50	77.05±3.63	78.81±2.29	71.81±3.85	69.65±8.77	86.67±4.67	81.73±9.05	74.22±3.88	73.82±6.26
	S	89.38±1.37	89.85±1.16	77.63±2.34	81.09±1.05	73.38±3.26	71.77±5.87	88.18±3.85	84.95±5.37	76.59±4.35	76.45±4.77
ANRMAB	–	86.52±1.91	85.90±2.91	73.06±5.15	73.29±5.02	68.43±7.41	67.47±6.03	87.53±2.48	87.41±2.46	77.93±3.01	79.12±2.25
	P	86.69±1.83	85.95±3.34	75.15±4.9	74.81±4.42	71.13±5.63	68.26±5.45	88.40±2.55	88.08±2.58	76.40±3.22	78.42±2.59
AGE	–	86.85±1.94	88.01±1.55	74.73±2.79	77.64±1.83	71.4±4.85	72.20±6.87	83.23±4.29	84.88±3.48	65.46±6.06	73.53±4.11
	P	86.70±1.56	88.17±1.27	73.84±1.74	77.69±1.59	72.20±4.79	72.37±6.03	83.34±3.92	85.49±3.84	65.37±7.56	73.63±3.76
GEEM	–	–	88.08±1.00	–	80.69±1.65	–	71.22±4.87	–	90.19±1.38	–	–
	P	–	86.74±1.89	–	78.92±2.97	–	71.01±3.59	–	90.04±1.27	–	–
MP	–	–	88.78±1.56	–	77.83±3.01	–	68.09±6.94	–	81.35±9.33	–	72.94±7.66
	P	–	86.89±2.92	–	81.79±1.62	–	72.01±4.41	–	83.29±5.35	–	75.55±7.45
ESP	–	–	83.47±2.36	–	81.55±1.77	–	71.72±5.43	–	89.87±1.29	–	–
	P	–	82.94±2.77	–	81.83±1.98	–	71.18±5.67	–	89.89±1.80	–	–

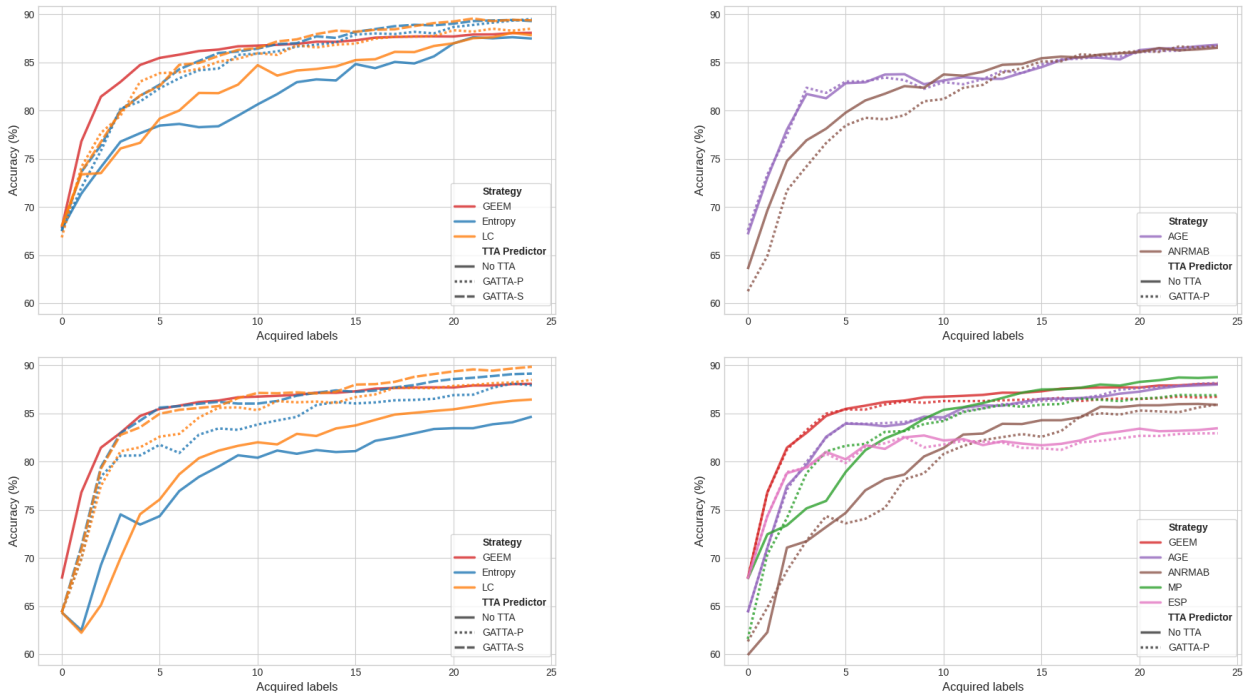

Figure 4: Active learning curves for Citeseer across acquisition strategies and architectures. The rows show performance for GCN (top) and SGC (bottom).

E Computational complexity

Figure 2 presents the computational overhead for a single active learning iteration across acquisition strategies, datasets, and GATTA variants. The runtime patterns remain consistent across datasets, revealing critical differences in computational scaling between aggregation approaches.

For simple strategies like Entropy, both GATTA-P and GATTA-S exhibit similar runtime scaling, with overhead growing modestly with the number of augmentations. This pattern extends to complex strategies when using GATTA-P, which maintains comparable runtime regardless of acquisition function complexity. In contrast, GATTA-S with complex strategies (e.g., AGE) exhibits substantially higher computational costs that scale linearly with both the number of augmentations and the acquisition function's complexity.

This disparity stems from the algorithmic differences. GATTA-P has computational complexity $\mathcal{O}((N \times I) + Q)$, where N is the number of augmentations, $\mathcal{O}(I)$ is the inference cost per augmentation, and $\mathcal{O}(Q)$ is the acquisition function evaluation cost. The acquisition function is computed only once on aggregated predictions. Conversely, GATTA-S has complexity $\mathcal{O}(N \times (I + Q))$, requiring $N + 1$ separate evaluations of the acquisition function—once per augmented view.

When $\mathcal{O}(Q)$ is negligible compared to $\mathcal{O}(I)$ (as with simple strategies like Entropy or LC), both variants exhibit similar runtimes. However, when $\mathcal{O}(Q)$ dominates, as with complex strategies like AGE that require expensive graph computations, GATTA-S incurs a multiplicative overhead of $N \times \mathcal{O}(Q)$, resulting in dramatically increased iteration times. For instance, at 10,000 augmentations with AGE on PubMed, GATTA-S requires over 5,000 seconds per iteration compared to approximately 50 seconds for GATTA-P, representing a 100-fold difference.

These results suggest a clear practical guideline: GATTA-P is preferable for complex acquisition functions due to its computational efficiency, while GATTA-S may be suitable for simple strategies where the additional repeated evaluations impose minimal overhead and potentially provide marginal performance benefits.

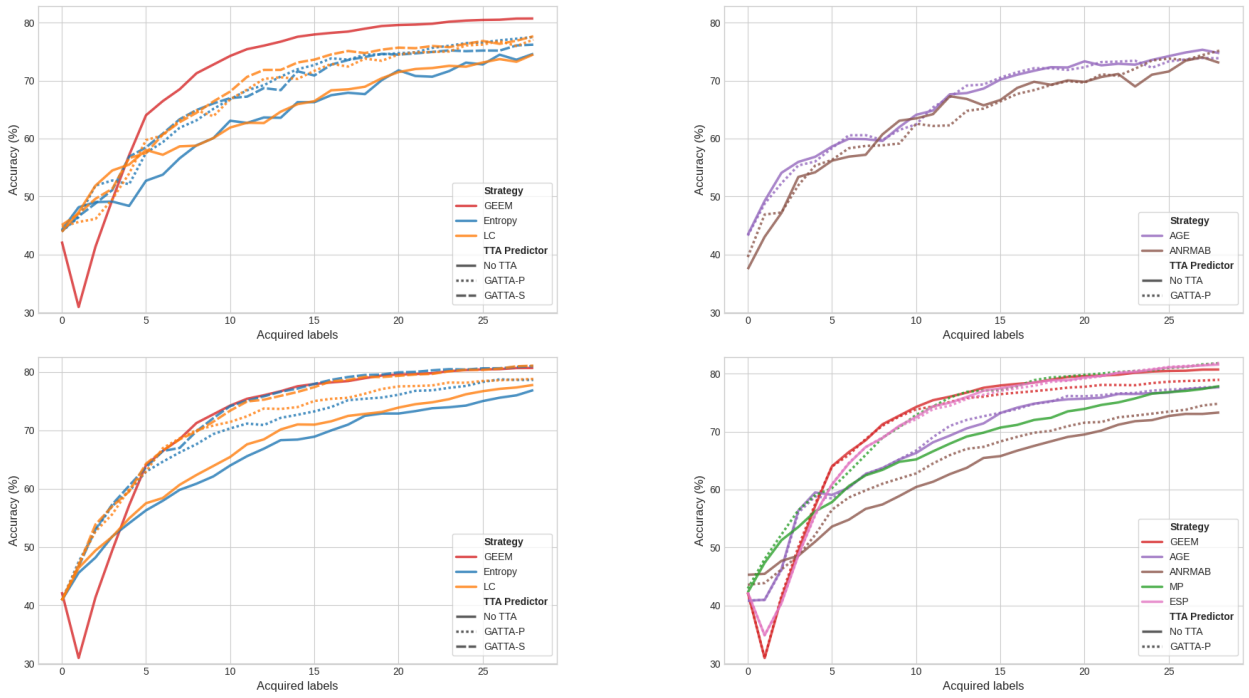

Figure 5: Active learning curves for CoraML across acquisition strategies and architectures. The rows show performance for GCN (top) and SGC (bottom).

F Confidence Analysis

Figure 9 illustrates the evolution of prediction confidence distributions across 25 active learning iterations for the Entropy strategy on Cora-ML with GATTA-P, averaged over 25 runs. Each panel displays a two-dimensional histogram where the y-axis represents confidence levels (0.0 to 1.0), the x-axis shows iteration number, color intensity indicates the frequency of nodes at each confidence level, and the red line traces mean confidence over iterations.

The baseline confidences exhibit a wide distribution throughout the active learning process. This broad dispersion reflects the inherent uncertainty in the model's predictions on unlabeled nodes. In contrast, GATTA confidences averaged over 500 augmented views (top right panel) show a more concentrated distribution, with increased density in the mid-range confidence region (0.4-0.6). This concentration effect suggests that test-time augmentation reveals underlying prediction uncertainty by reducing both overconfident and underconfident predictions, moderating them toward more calibrated estimates.

Notably, the mean confidence remains consistently above the most concentrated region in both the original and GATTA distributions, indicating that the distribution is skewed toward higher confidences. This asymmetry is characteristic of uncertainty-based active learning, where the strategy progressively queries uncertain nodes, leaving a larger proportion of high-confidence predictions in the unlabeled pool.

The filtered GATTA confidences demonstrate the effect of consistency-based filtering. Compared to unfiltered GATTA, filtering shifts the distribution upward, reducing the density of low-confidence predictions (<0.4). This elevation effect occurs because filtering discards augmented views with inconsistent predictions, retaining only the more stable, higher-confidence predictions. The result is a distribution that is both more concentrated and shifted toward higher confidence values, suggesting improved calibration through the removal of unreliable augmentation-induced predictions.

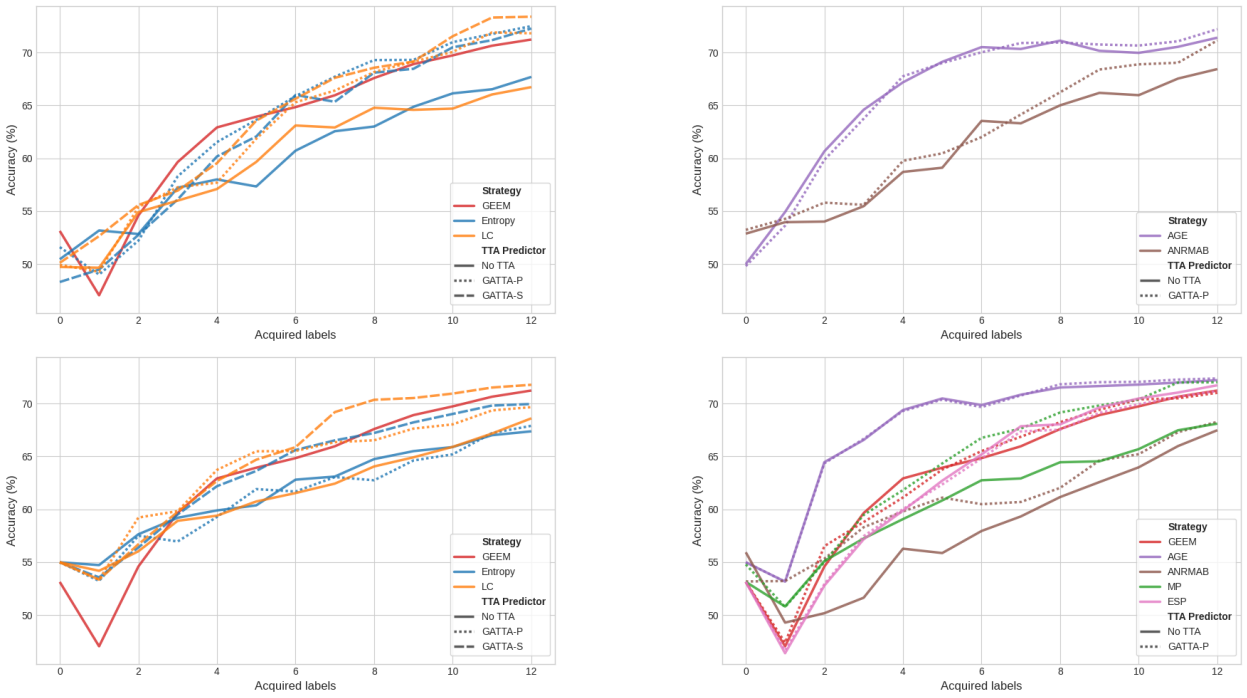

Figure 6: Active learning curves for PubMed across acquisition strategies and architectures. The rows show performance for GCN (top) and SGC (bottom).

These distributional changes have direct implications for active learning: the concentration and elevation effects indicate that GATTA provides more reliable uncertainty estimates, potentially leading to better node selection and improved label efficiency.

Table 8: Comparison of GATTA variants with MC Dropout (MCD) on Entropy-based acquisition. GATTA-P and GATTA-S consistently match or outperform MCD, while their combination yields no additive benefit. Best results per dataset in bold.

	Amazon Computers	Amazon Photos	CiteSeer	CoraML	PubMed
Entropy	70.06 ± 7.97	84.47 ± 6.12	87.48 ± 3.44	74.61 ± 4.7	67.69 ± 6.13
+ MCD	**75.77 ± 3.64**	83.98 ± 4.41	88.8 ± 1.72	76.54 ± 3.62	68.76 ± 5.63
GATTA-P	70.04 ± 7.05	84.8 ± 6.13	88.61 ± 1.75	**77.5 ± 2.72**	**72.49 ± 3.6**
+ MCD	61.67 ± 9.31	83.49 ± 5.83	**89.51 ± 1.07**	75.06 ± 3.11	69.69 ± 5.61
GATTA-S	68.54 ± 10.98	**86.67 ± 3.91**	89.35 ± 1.01	76.7 ± 2.83	71.39 ± 4.58
+ MCD	74.3 ± 5.5	84.17 ± 7.42	89.26 ± 1.45	75.97 ± 2.74	70.54 ± 4.74

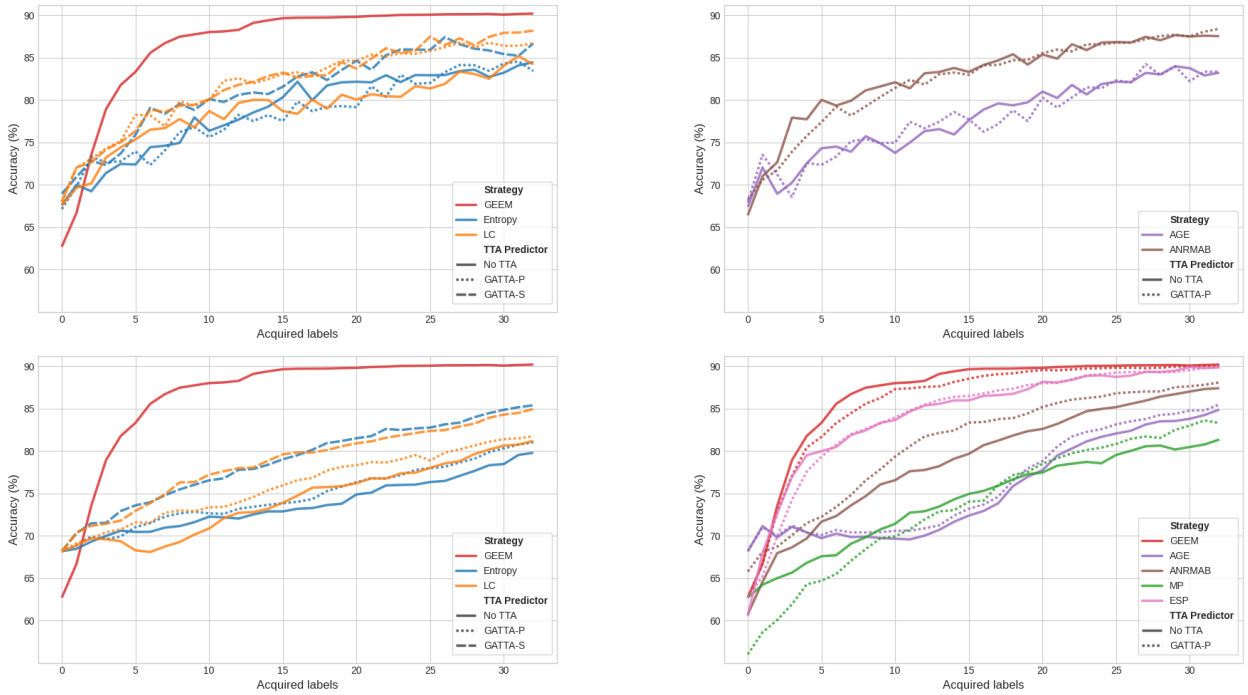

Figure 7: Active learning curves for Amazon Photos across acquisition strategies and architectures. The rows show performance for GCN (top) and SGC (bottom).

Table 9: GATTA performance across GNN architectures with Entropy strategy. Rows marked "–" denote baseline (no GATTA), "P" denotes GATTA-P with filtering, and "S" denotes GATTA-S without filtering. All four architectures show consistent improvements over baselines, confirming GATTA's architecture-agnostic design. Best results per architecture and dataset in bold.

		Amazon Computers	Amazon Photos	CiteSeer	CoraML	PubMed
GCN	–	70.06 ± 7.81	84.47 ± 5.99	87.48 ± 3.37	74.61 ± 4.61	67.69 ± 6.01
	P	70.04 ± 6.91	83.41 ± 6.34	**89.51 ± 1.60**	**77.50 ± 2.67**	**72.49 ± 3.53**
	S	**74.58 ± 6.51**	**86.67 ± 3.83**	89.30 ± 1.35	76.19 ± 3.35	72.27 ± 5.49
SGC	–	69.18 ± 6.63	79.79 ± 8.91	84.67 ± 6.29	76.89 ± 4.06	67.37 ± 5.79
	P	70.08 ± 7.24	81.02 ± 7.98	87.93 ± 1.81	78.64 ± 2.83	67.89 ± 7.36
	S	**74.57 ± 4.2**	**85.39 ± 5.99**	**89.14 ± 1.22**	**81.01 ± 1.63**	**69.95 ± 6.75**
GAT	–	72.22 ± 2.10	82.41 ± 1.85	86.97 ± 4.61	73.40 ± 5.16	66.16 ± 7.77
	P	75.46 ± 2.23	85.46 ± 1.74	89.80 ± 4.03	74.14 ± 5.17	**69.97 ± 5.87**
	S	**77.61 ± 1.33**	**85.73 ± 2.02**	**90.02 ± 3.4**	**75.96 ± 5.12**	67.87 ± 5.41
GraphSAGE	–	60.91 ± 3.02	78.05 ± 3.59	85.31 ± 6.47	71.84 ± 7.85	**63.98 ± 8.05**
	P	**63.09 ± 2.86**	77.86 ± 5.78	84.96 ± 6.29	69.23 ± 7.57	63.26 ± 3.31
	S	62.84 ± 1.93	**79.61 ± 4.62**	**86.97 ± 5.38**	**72.25 ± 12.93**	62.77 ± 4.34

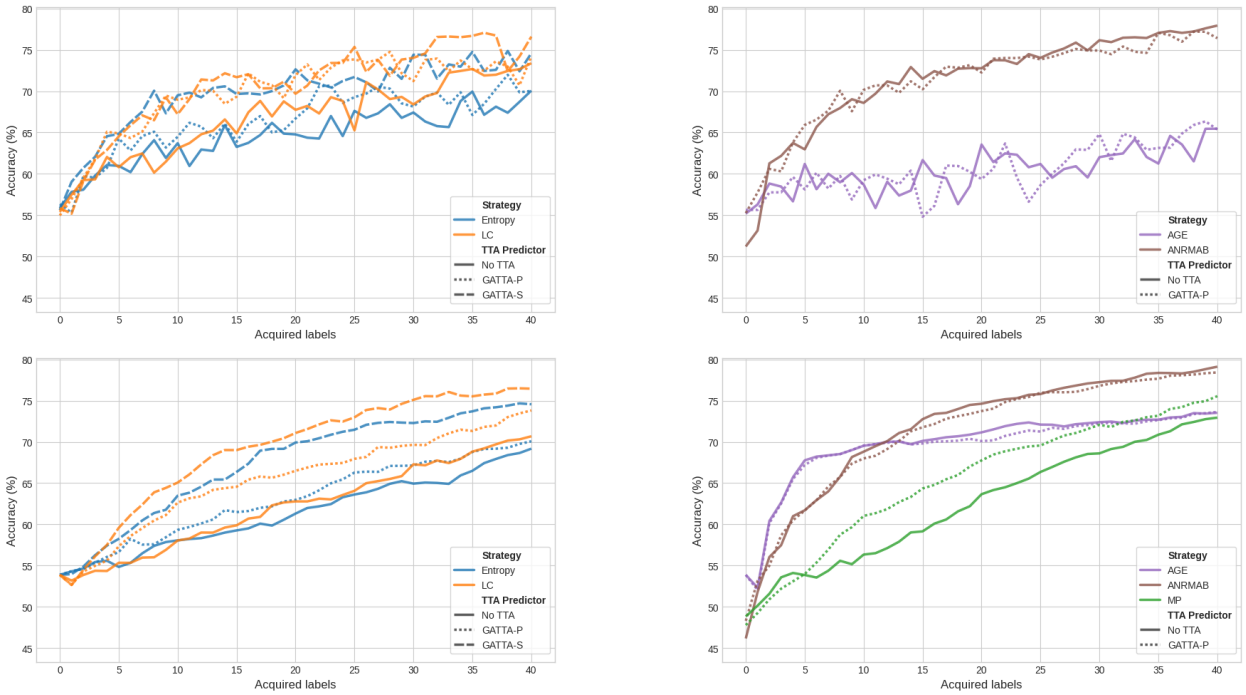

Figure 8: Active learning curves for Amazon Computers across acquisition strategies and architectures. The rows show performance for GCN (top) and SGC (bottom).

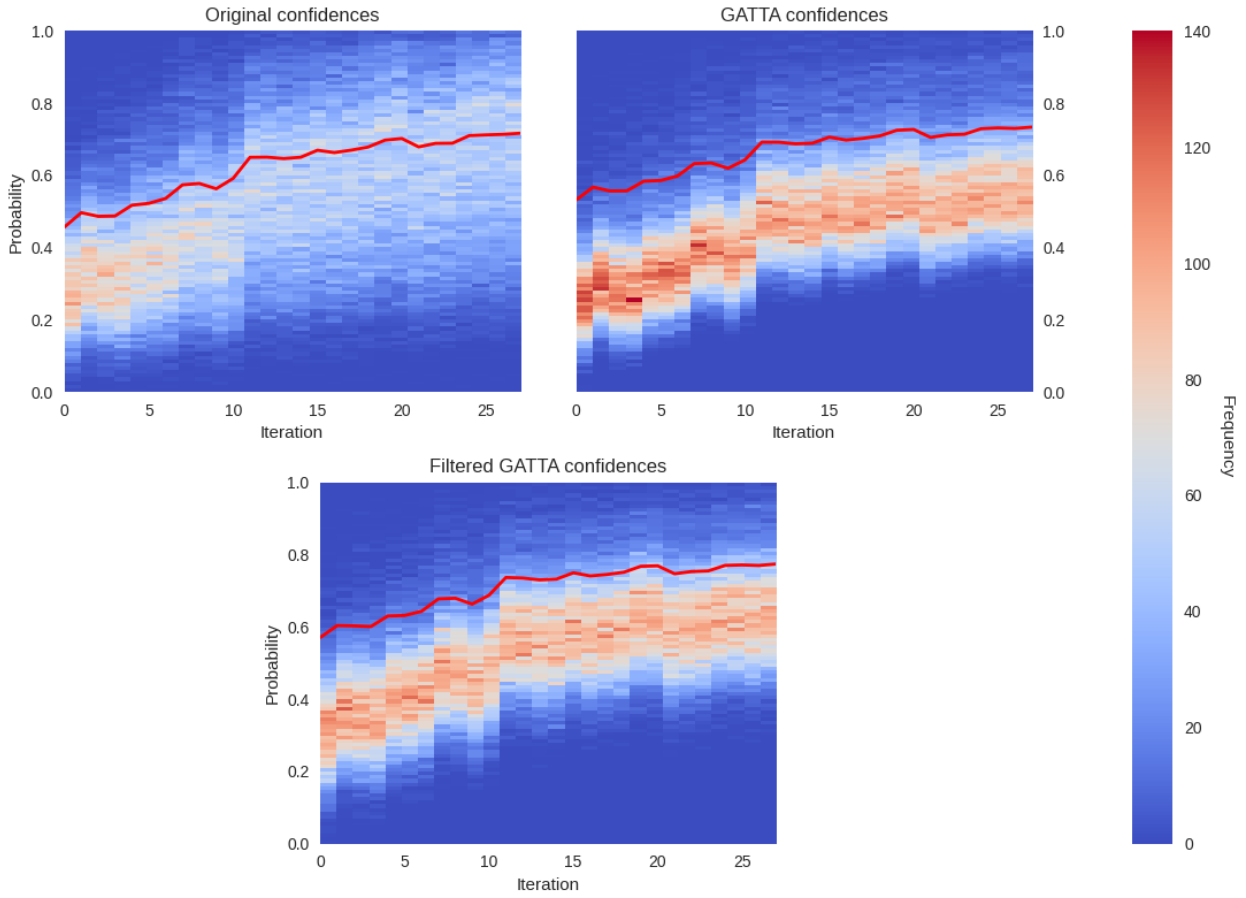

Figure 9: Confidence distribution evolution across active learning iterations for Entropy on Cora-ML with GATTA-P. Heat maps show the distribution of prediction confidences (y-axis) over active learning iterations (x-axis), averaged over 25 runs. Top left: baseline confidences from the original graph. Top right: confidences averaged over 500 augmented views. Bottom: confidences after applying consistency-based filtering. The red line indicates mean confidence per iteration. Color intensity represents the frequency of nodes at each confidence level.

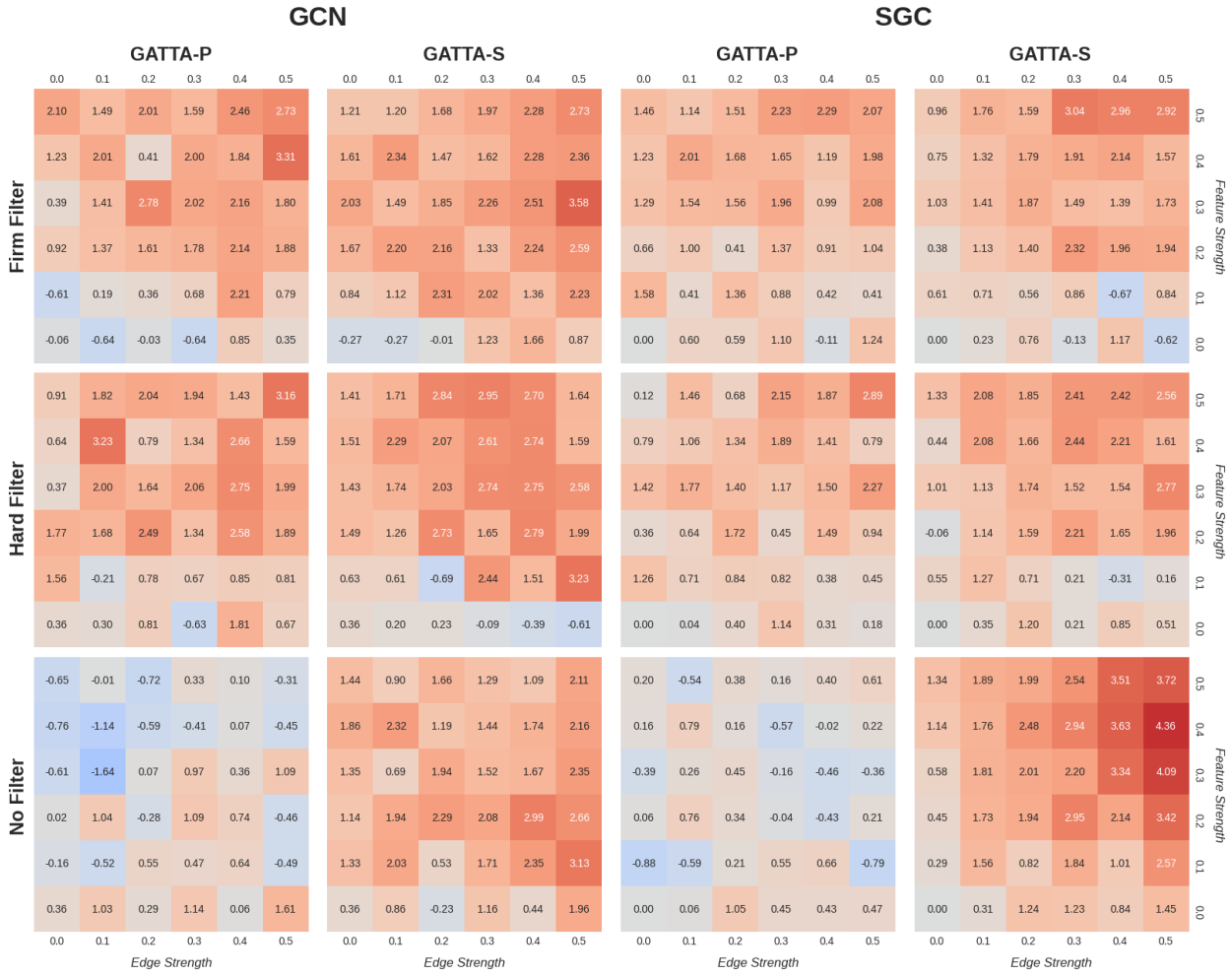

Figure 10: Performance sensitivity to augmentation strength and filtering for **Entropy** on **CoraML**. Heatmaps show performance improvement (%) relative to baseline for GATTA-P and GATTA-S with GCN and SGC architectures. Rows correspond to Firm Filter, Hard Filter, and No Filter. Axes represent edge dropout (horizontal) and feature noising (vertical) strengths from 0.0 to 0.5. Darker red indicates larger improvements; blue indicates degradation.

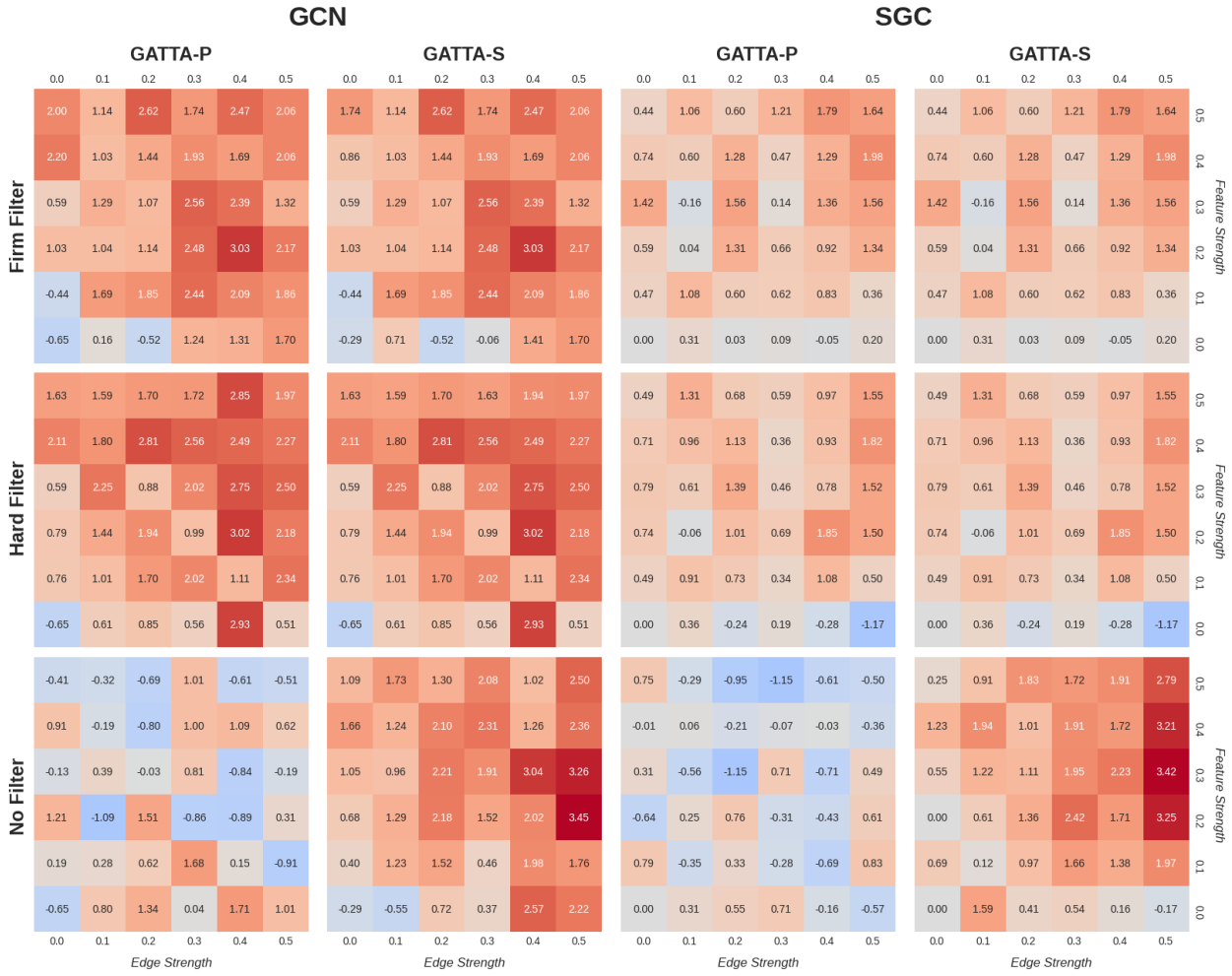

Figure 11: Performance sensitivity to augmentation strength and filtering for **LC** on **CoraML**. Heatmaps show performance improvement (%) relative to baseline for GATTA-P and GATTA-S with GCN and SGC architectures. Rows correspond to Firm Filter, Hard Filter, and No Filter. Axes represent edge dropout (horizontal) and feature noising (vertical) strengths from 0.0 to 0.5. Darker red indicates larger improvements; blue indicates degradation.

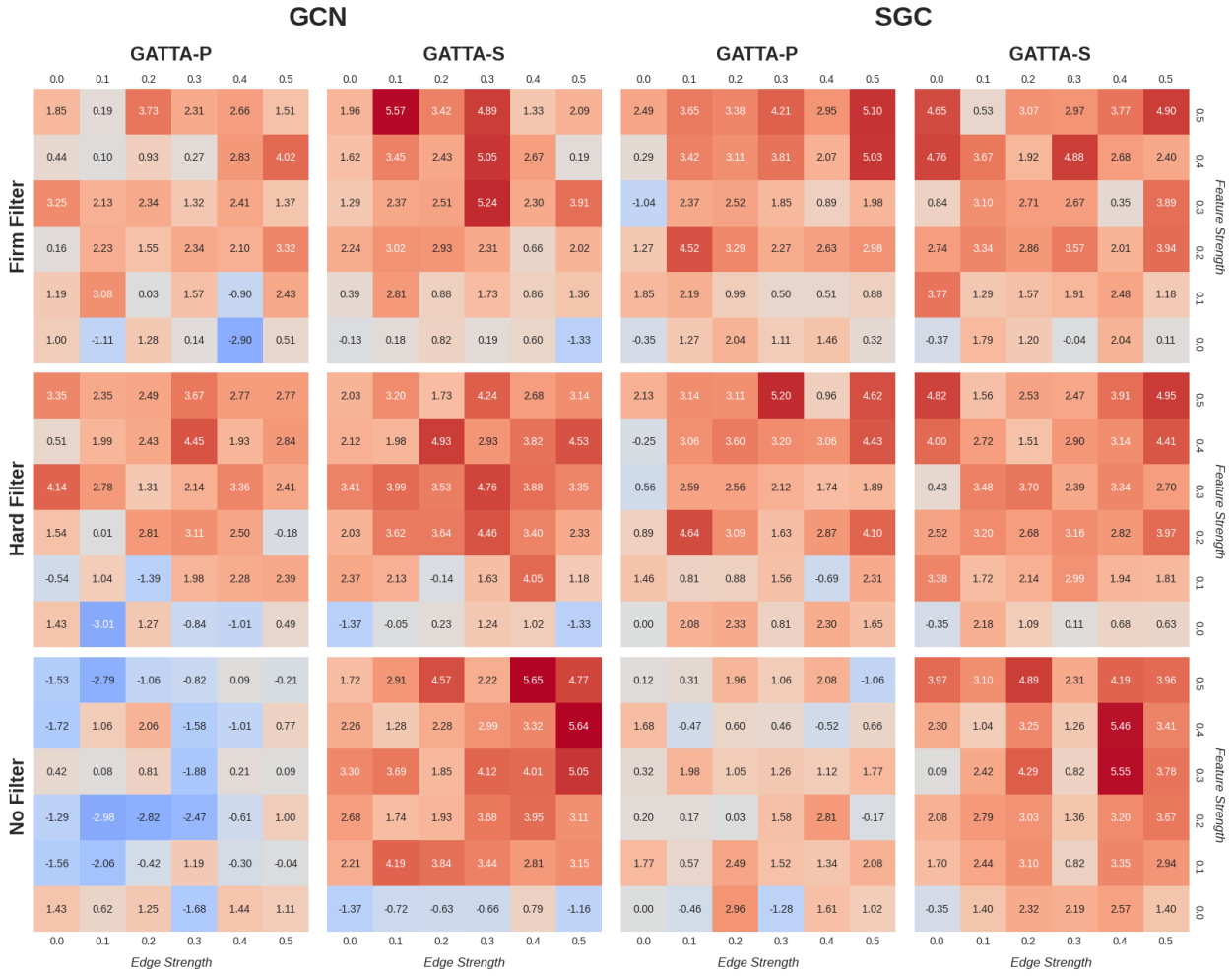

Figure 12: Performance sensitivity to augmentation strength and filtering for **Entropy** on **PubMed**. Heatmaps show performance improvement (%) relative to baseline for GATTA-P and GATTA-S with GCN and SGC architectures. Rows correspond to Firm Filter, Hard Filter, and No Filter. Axes represent edge dropout (horizontal) and feature noising (vertical) strengths from 0.0 to 0.5. Darker red indicates larger improvements; blue indicates degradation.

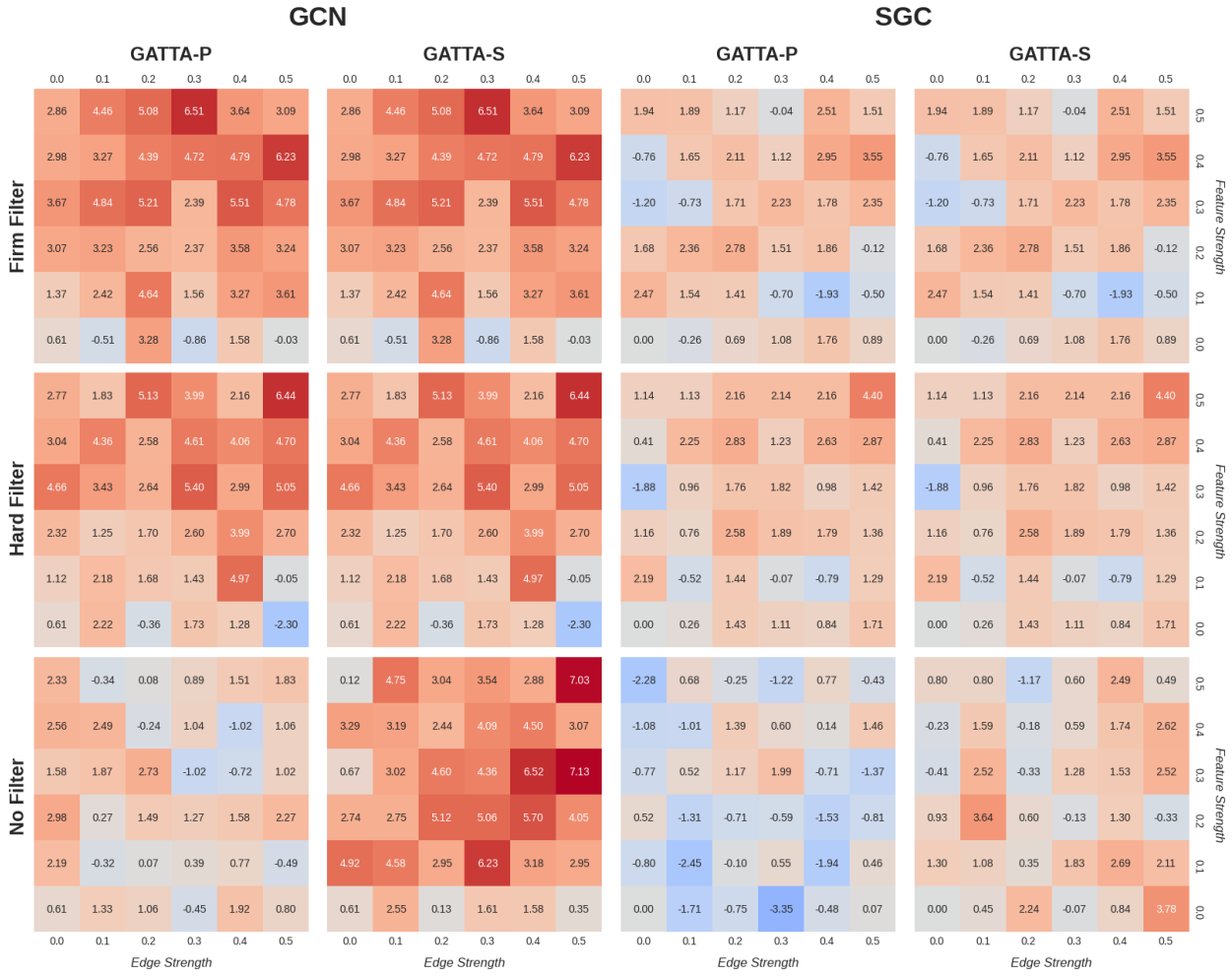

Figure 13: Performance sensitivity to augmentation strength and filtering for **LC** on **PubMed**. Heatmaps show performance improvement (%) relative to baseline for GATTA-P and GATTA-S with GCN and SGC architectures. Rows correspond to Firm Filter, Hard Filter, and No Filter. Axes represent edge dropout (horizontal) and feature noising (vertical) strengths from 0.0 to 0.5. Darker red indicates larger improvements; blue indicates degradation.