WHETHER TO TRUST GNNS OR LLMS? FAIRROUTER FOR FEW-SHOT NODE CLASSIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Text-attributed graphs (TAGs) couple textual semantics with graph topology. Recent GNN–LLM collaborative approaches exploit these complementary modalities to generate pseudo-labels for few-shot learning. However, existing approaches typically fix one model as the annotator or judge, regardless of which modality is more reliable for a given node, leading to biased pseudo-labels and underutilized complementary information. To this end, we propose **FairRouter**, which generates pseudo-labels through fair modality routing. Specifically, the router characterizes each node with multi-level evidence from both models and learns two signals: a trust score that decides whether the node yields a reliable pseudo-label, and a preference score that decides which modality to follow when the two disagree. To generalize from scarce labels, the router learns from augmented views, where semantic and topological perturbations are applied to simulate diverse modality trust and preference patterns. The routed pseudo-labels then supervise a multimodal residual classifier, which recovers nodes misclassified by both modalities while preserving their consensus. Experiments on five TAG benchmarks show that FairRouter achieves the best overall accuracy under all label budgets and the highest pseudo-label precision on all datasets, with gains of up to nine percentage points in the most label-scarce settings. The codebase is publicly available[1].

1 INTRODUCTION

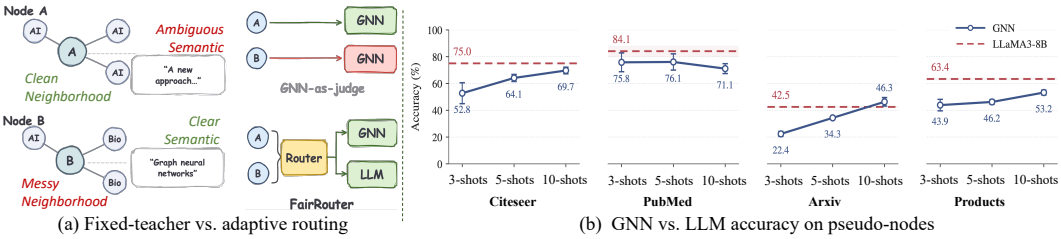
Text-attributed graphs (TAGs), in which nodes carry rich textual descriptions, are ubiquitous in citation networks, social media, e-commerce, and knowledge graphs (Chang & Blei, 2009; Yang et al., 2015; Hu et al., 2020; Zhao et al., 2025; Tang et al., 2025). Learning on TAGs requires integrating node-level textual semantics with inter-node structural dependencies (Wang et al., 2025; Liu et al., 2026b). Graph neural networks (GNNs) excel at encoding structure through neighborhood aggregation (Kipf & Welling, 2017; Wu et al., 2019), whereas large language models (LLMs) offer strong semantic understanding (Grattafiori et al., 2024; Yang et al., 2025). Their complementary strengths have motivated GNN-LLM collaborative frameworks that either equip LLMs with graph-aware context (Tang et al., 2024; Chen et al., 2024a) or enable bidirectional knowledge transfer (Feng et al., 2025; Loveland et al., 2026).

In practice, however, TAGs are often sparsely labeled due to high annotation costs (Luo et al., 2024; Yu et al., 2025; Wang et al., 2025), and learning decision boundaries that generalize across the graph from only a few labeled nodes remains difficult (Li et al., 2018; 2023b; Wang et al., 2024; Wei et al., 2026). Pseudo-labeling alleviates this scarcity by exploiting abundant unlabeled nodes (Rizve et al., 2021; Shao et al., 2026). Recent GNN–LLM collaborative pseudo-labeling methods (Chen et al., 2024b; Pei et al., 2024; Sheng et al., 2025; Zhang et al., 2025; Xu & Ding, 2026; Liu et al., 2026a) typically let one model generate candidate labels while the other selects nodes, propagates labels, or filters noisy annotations. For instance, GNN-as-Judge (Xu & Ding, 2026) employs GNN as a topology-aware judge to produce reliable pseudo-labels on selected influential nodes, thereby enhancing LLMs for few-shot classification.

However, modality reliability is inherently node-dependent (Mao et al., 2023; Zhang et al., 2023). As illustrated in Figure 1(a), GNNs are trustworthy in well-connected, homophilous regions but degrade under noisy neighborhoods, whereas LLMs excel on nodes with informative text but fail when textual descriptions or class semantics are ambiguous. Existing methods overlook this variation and

[1]The code of our project is available at https://anonymous.4open.science/r/FairRouter-C6F1.

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.



(a) Fixed-teacher vs. adaptive routing (b) GNN vs. LLM accuracy on pseudo-nodes

Figure 1: Motivation for node-wise routing. (a) Fixed-teacher pseudo-labeling (e.g., GNN-as-Judge) always favors the GNN, whereas FairRouter decides per node which modality to trust. (b) On nodes selected by GNN-as-Judge, the LLM outperforms the GNN in most settings, showing that a fixed GNN preference is biased.

fix the roles of the two models in advance, treating one as the dominant annotator or judge. GNN-as-Judge, for example, always adopts the GNN prediction when the two models disagree; yet on the very nodes it selects, the LLM is more accurate than the GNN in most settings (Figure 1(b)). Such a globally fixed preference propagates the errors of the favored model into pseudo-labels while under-exploiting the other. Therefore pseudo-labeling should be guided by *node-level* modality reliability, which raises two challenges:

❶ *How to sense node-wise modality reliability?* Pseudo-labeling under node-dependent reliability amounts to a node-wise routing decision: which modality, if any, to trust. Yet modality quality is neither directly observable nor comparable (Ong et al., 2025; Loveland et al., 2026): single-modality confidence is biased and ignores complementary cues (Liu et al., 2022; Sui & Hooi, 2026), while heterogeneous models differ in architecture, input space, and calibration, rendering their outputs incomparable (Guo et al., 2017; Wang et al., 2021; Xiong et al., 2024). Reliable routing thus requires assessing cross-modal evidence in a shared space.

❷ *How to learn beyond routed pseudo-labels?* Routing only selects among existing predictions and cannot recover nodes on which both modalities fail. Moreover, routed pseudo-labels are unevenly distributed across modalities, allowing the more frequently selected one to dominate a jointly trained classifier (Jiang et al., 2026; Gong et al., 2026), and naive error correction risks overwriting reliable consensus where both modalities agree (Ricci et al., 2026; Wei et al., 2020). A unified classifier is thus needed to fuse both modalities under balanced supervision, correcting shared errors while preserving consensus.

To address these challenges, we propose **FairRouter**, a few-shot learning framework for TAGs that generates pseudo-labels via *fair* modality routing: rather than predefining a dominant teacher, it adaptively decides, for each unlabeled node, which modality to trust. *For Challenge 1*, a learnable router integrates multi-level evidence from both modalities into node-wise trust and preference scores, which identify reliable pseudo-labels and resolve inter-modality conflicts. To learn transferable reliability patterns from scarce labels, the router is trained on augmented views with perturbed modality preferences, enabling generalization to unseen nodes. *For Challenge 2*, a multimodal residual classifier fuses dual-model predictions with a cross-modal residual that rectifies errors shared by both models, and is guided by modality-balanced supervision and a consensus-preserving constraint. Our main contributions are summarized as follows:

- We propose FairRouter, a node-adaptive modality routing framework for fair and reliable pseudo-labeling on few-shot TAGs, which addresses the bias of fixed-teacher methods under varying modality reliability.
- We design a learnable router that leverages multiple augmented views to capture diverse modality trust and preference patterns, enabling dynamic quality sensing and global generalization.
- We introduce a multimodal residual classifier that goes beyond modality selection, recovering nodes misclassified by both modalities while preserving their consensus.

2 Related Work

Existing studies have extensively explored pseudo-labeling to augment low-resource learning on text-attributed graphs (TAGs). Based on how structural and semantic modalities are leveraged for pseudo-label generation, existing approaches can be broadly categorized into three lines of research.

GNN-based pseudo-labeling. Early self-training approaches on graphs primarily rely on GNNs themselves to generate pseudo-labels. For example, Li et al. (2023a), GLIM (Hu et al., 2023), MoDis (Pei et al., 2024), and TA-GST (Liu et al., 2026a) improve pseudo-label quality from the perspectives of prediction confidence, label distribution, training dynamics, and topological consistency, respectively. These studies underscore that reliable pseudo-label selection is critical to the success of few-shot learning. However, the resulting pseudo-labels remain susceptible to the GNN's inherent biases and fail to fully exploit rich semantic features in TAGs.

LLM-based pseudo-labeling. Recent studies increasingly leverage LLMs as pseudo-label sources for TAG learning, benefiting from their strong semantic reasoning and understanding capabilities. LLM-GNN (Chen et al., 2024b) actively selects representative and diverse nodes for LLM annotation, which are then used for GNN training. Li et al. (2024) further identifies nodes where GNN is unreliable but LLM is competent, and distills LLM's pseudo-labels, predictive distributions, and reasoning signals into GNN. More recently, LLM4NG (Yu et al., 2025) prompts an LLM to generate class-related synthetic nodes and connects them through an edge predictor, implicitly enriching supervision under few-shot settings, while DMA (Sheng et al., 2025) estimates class-wise noise distribution and incorporates it into graph active learning to prioritize nodes that are more informative and reliable for LLM annotation. Additionally, PKD (Wei et al., 2026) introduces preference-driven distillation to improve the transfer of knowledge from LLMs to GNNs. These approaches follow an LLM-to-GNN paradigm, where the LLM provides pseudo supervision to improve GNN. Consequently, their effectiveness depends heavily on the quality of LLMs, while making it difficult to leverage structural modality to complement text-based judgments.

GNN–LLM Collaborative pseudo-labeling. More recent studies have begun to explicitly consider the complementary strengths of GNNs and LLMs during pseudo-label generation. Locle (Zhang et al., 2025) adopts an active self-training framework, iteratively selecting critical nodes via GNNs and refines their pseudo-labels by combining LLM predictions over a rewired topology. In contrast, GNN-as-Judge (Xu & Ding, 2026) explicitly exploits disagreement cases between GNN and LLM and assigns the GNN predictions as positive to construct preference pairs for training, establishing a collaborative mechanism in which the GNN acts as the principal decision maker. Nevertheless, existing approaches still impose an asymmetric role on GNNs and LLMs: they typically favor a single modality by fixing one model as the annotator or judge, regardless of which modality is more reliable for different nodes and graph regions.

To sum up, our approach aims to treat GNNs and LLMs **fair**ly and sense their relative strengths dynamically to **route** pseudo-label generation to the more reliable modality on different TAG regions.

3 Proposed Approach

We propose **FairRouter**, a multi-modal pseudo-generation framework for few-shot learning on text-attributed graphs. As illustrated in Figure 2, it consists of two stages: (i) *Quality-Aware Fair Routing*, which estimates node-wise trust and preference scores and routes trustworthy nodes to reliable pseudo-labels, and (ii) *Multi-modal Residual Classifier*, which leverages trustworthy pseudo-labels to guide residual correction and adaptive fusion of the two modalities, thereby resolving all query nodes.

3.1 Preliminary

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ be a text-attributed graph, where $\mathcal{V}$ and $\mathcal{E}$ are nodes and edges, and $\mathcal{X} = \{x_i\}_{i\in\mathcal{V}}$ denotes node texts. Under the k-shot setting, each class $c \in \{1, \dots, C\}$ has $\min(|\mathcal{V}_c|, k)$ labeled nodes, where $\mathcal{V}_c$ is the set of nodes in class c; these form the support set $\mathcal{S}$, and the remaining nodes $\mathcal{U} = \mathcal{V} \setminus \mathcal{S}$ are unlabeled. We consider two pretrained, frozen models: a GNN F_G and an LLM F_L, which produce class distributions p_i^E, representations h_i^E, and predictions $\hat{y}_i^E = \arg\max_c p_i^E(c)$ for $E \in \{G, L\}$. We partition $\mathcal{V}$ into agreement nodes $\mathcal{V}_A$, where $\hat{y}_i^G = \hat{y}_i^L$, and disagreement nodes $\mathcal{V}_D$, where they differ.

3.2 Quality-Aware Fair Routing

We formulate pseudo-label generation as a fair routing problem to sense the quality for different modalities, in which a router fairly exploits joint *multi-modal evidence* to decide *whether to trust* the predictions and *which modality to prefer*. To acquire node-level awareness from limited supervision, we further expose the router to diverse model behaviors through *multi-view augmentation*.

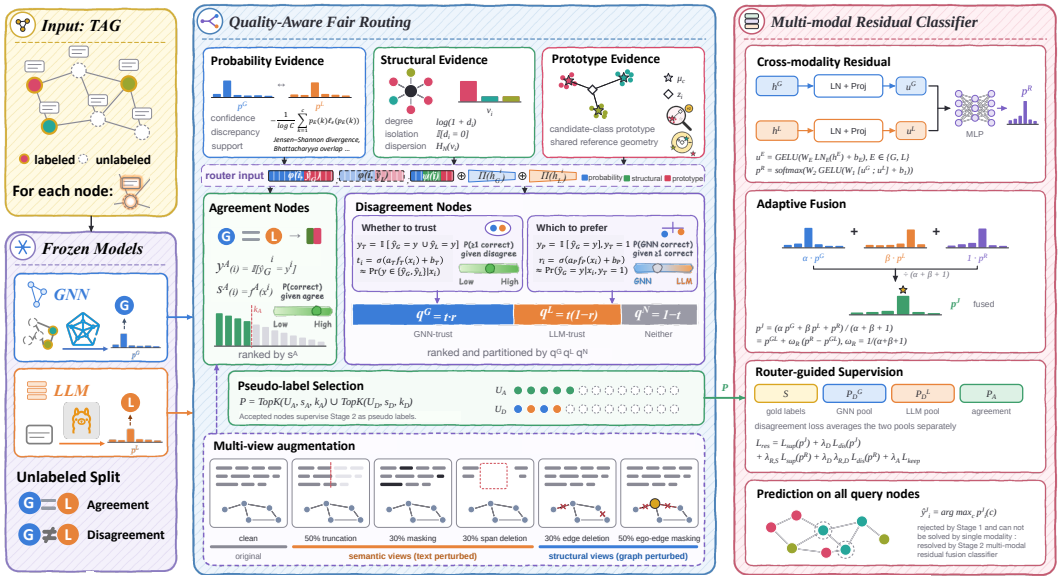


Figure 2: Framework of **FairRouter** for few-shot semi-supervised node classification on TAGs.

Multi-modal Evidence. For each candidate prediction $c \in \{\hat{y}_i^G, \hat{y}_i^L\}$, we construct a candidate-specific evidence vector $\phi(i, c)$ and a candidate-agnostic descriptor $\psi(i)$ from three extractors: (i) *probability extractor*, capturing individual confidence, inter-modality discrepancy, and candidate-specific support; (ii) *Neighborhood extractor*, capturing node degree, isolation, and neighborhood prediction dispersion without requiring true labels; and (iii) *prototype extractor*, which compares each node with labeled prototypes constructed from few-shot references in the frozen representation space. The explicit evidence is $[\phi(i, c); \psi(i)]$ for agreement and $[\phi(i, \hat{y}_i^G); \phi(i, \hat{y}_i^L); \psi(i)]$ for disagreement nodes. It is further concatenated with projections of h_i^G and h_i^L to form router input $\mathbf{x}_i$. More details are in Appendix A.

Whether to Trust. Agreement and disagreement nodes exhibit different reliability patterns. For *agreement* nodes, the shared prediction may still be incorrect; we therefore train a scorer f_A with the binary target $y_A(i) = \mathbb{I}[\hat{y}_i^G = y_i]$ and obtain $s_A(i) = f_A(\mathbf{x}_i)$, which is used only for ranking and need not be calibrated. For *disagreement* nodes, we ask whether candidates from the two modalities contain a correct answer, with the target

$$y_T(i) = \mathbb{I}\big[\hat{y}_i^G = y_i \ \vee \ \hat{y}_i^L = y_i\big], \qquad i \in \mathcal{U}_D. \tag{1}$$

A trust scorer f_T is trained on y_T, and its output is mapped to

$$t_i = \sigma(a_T f_T(\mathbf{x}_i) + b_T) \approx \Pr\big(y_i \in \{\hat{y}_i^G, \hat{y}_i^L\} \mid \mathbf{x}_i,\ i \in \mathcal{U}_D\big). \tag{2}$$

The corresponding implementation details are provided in Appendix B.1 and B.2.

Which to Prefer. Conditioned on $y_T(i) = 1$, the candidate set of a disagreement node contains the correct answer, modality selection reduces to a binary decision. We therefore model preference as the conditional probability that the GNN is correct given $y_T(i) = 1$. We train a preference classifier f_P with the target $y_P(i) = \mathbb{I}[\hat{y}_i^G = y_i]$, which yields $r_i \approx \Pr(\hat{y}_i^G = y_i \mid \mathbf{x}_i,\ i \in \mathcal{U}_D,\ y_T(i) = 1)$; observations in which neither modality is correct are excluded from this task. Trust and preference are combined into three state scores

$$q_G(i) = t_i r_i, \qquad q_L(i) = t_i(1 - r_i), \qquad q_N(i) = 1 - t_i, \tag{3}$$

which partitions disagreement nodes into *GNN-trust*, *LLM-trust*, and *None* regions. This factorization follows the conditional probability rule without assuming independent modality errors and ensures that a strong preference for one modality yields a high score only when at least one modality is trustworthy.

Pseudo-Label Selection. For each disagreement node, we select $e_i^\star = \arg\max_{E \in \{G,L\}} q_E(i)$, assign $\tilde{y}_i = \hat{y}_i^{e_i^\star}$, and score it by $s_D(i) = q_{e_i^\star}(i)$. As the two branches differ in reliability patterns and score scales, they are ranked separately, and the accepted set is

$$\mathcal{P} = \mathrm{TopK}(\mathcal{U}_A, s_A, k_A) \cup \mathrm{TopK}(\mathcal{U}_D, s_D, k_D), \tag{4}$$

where $\mathrm{TopK}(\mathcal{U}, s, k)$ returns the K highest-scoring nodes and k_A, k_D are branch-specific budgets. More details can be seen in Appendix B.3.

Multi-View Augmentation. To expose the router to diverse model behaviors without additional labels, we construct semantic and structural perturbations on the support set and recompute predictions using the frozen models in each augmented environment. Every observation retains its gold label, whereas its branch membership and routing targets y_A, y_T, y_P are recomputed from the environment-specific predictions. Different views of the same node can thus supervise different routing tasks, while the models remain frozen. More details are in Appendix B.4.

3.3 Multi-modal Residual Classifier

We learn a unified classifier on the complementary representations and fuse its residual output with dual-modality distributions, leveraging the router-accepted pseudo-labels as additional supervision.

Cross-modality Residual. We project h_i^G and h_i^L into a shared d_R-dimensional space through separate normalization and projection layers, and feed the concatenated projections $[u_i^G; u_i^L]$ into a shared MLP to obtain a residual distribution p_i^R over all C classes. The residual classifier can therefore propose a class beyond the top-one predictions from both modalities (Appendix C.1).

Adaptive Fusion Classifier. We fuse the original distributions with two learnable positive coefficients $\alpha = \mathrm{softplus}(a_G)$ and $\beta = \mathrm{softplus}(a_L)$, both initialized to one:

$$p_i^J = \frac{\alpha p_i^G + \beta p_i^L + p_i^R}{\alpha + \beta + 1} = p_i^{GL} + \omega_R \left(p_i^R - p_i^{GL}\right), \tag{5}$$

where $p_i^{GL} = (\alpha p_i^G + \beta p_i^L)/(\alpha + \beta)$ is the multi-modal mixture and $\omega_R = 1/(\alpha + \beta + 1)$. The residual classifier thus provides a node-dependent correction to the multi-modal mixture while preserving a normalized class distribution, and the final prediction is $\hat{y}_i^J = \arg\max_c p_i^J(c)$.

Router-Guided Supervision. Let $\mathcal{P}_A = \mathcal{P} \cap \mathcal{U}_A$, and let $\mathcal{P}_D^E$ denote the accepted disagreement nodes whose selected modality is $E \in \{G, L\}$. The support loss $\mathcal{L}_{\mathrm{sup}}$ is the cross-entropy on $\mathcal{S}$. The disagreement loss $\mathcal{L}_{\mathrm{dis}}$ averages the cross-entropy over $\mathcal{P}_D^G$ and $\mathcal{P}_D^L$ separately, so that a larger pool cannot dominate through its size. On $\mathcal{P}_A$, $\mathcal{L}_{\mathrm{keep}}$ keeps p^J close to the mean distribution $\frac{1}{2}(p_i^G + p_i^L)$, discouraging unnecessary changes to the consensus. The overall objective is

$$\mathcal{L}_{\mathrm{residual}} = \mathcal{L}_{\mathrm{sup}}(p^J) + \lambda_D \mathcal{L}_{\mathrm{dis}}(p^J) + \lambda_{R,S} \mathcal{L}_{\mathrm{sup}}(p^R) + \lambda_D \lambda_{R,D} \mathcal{L}_{\mathrm{dis}}(p^R) + \lambda_A \mathcal{L}_{\mathrm{keep}}, \tag{6}$$

where all coefficients are nonnegative (Appendix C.2). At inference, Eq. 5 is applied to every query node, including those excluded from pseudo-label supervision.

Overall, FairRouter progresses from identifying trustworthy modalities to collaborative multi-modal learning: the router converts modality reliability into pseudo-supervision, and the residual classifier turns this supervision into cross-modality fusion and complementary correction.

4 Experiments

4.1 Experimental Setup

Datasets. We use five TAG benchmarks: `Cora` (Yang et al., 2016), `Citeseer`, `PubMed` (Sen et al., 2008), and two large-scale OGB graphs (Hu et al., 2020), ogbn-arxiv (Arxiv) and the ogbn-products subset (Products). Following GNN-as-Judge (Xu & Ding, 2026), we adopt its data splits and sample $k \in \{3, 5, 10\}$ labeled nodes per class. The heterophilous Cornell and Wisconsin (Craven et al., 1998; Pei et al., 2020) are used to test generalization. Dataset statistics are given in Appendix D.1.

Baselines. We compare with (i) *GNNs*: GCN (Kipf & Welling, 2017) and SGC (Wu et al., 2019); (ii) *graph-LLM models*: GLEM (Zhao et al., 2023), LLaGA (Chen et al., 2024a), and GraphGPT (Tang et al., 2024); and (iii) *pseudo-labeling methods* from all three paradigms in Section 2: GNN self-training (MoDis (Pei et al., 2024), TA-GST (Liu et al., 2026a)); LLM-to-GNN annotation (LLM-GNN (Chen et al., 2024b), DMA (Sheng et al., 2025)); and GNN-LLM collaboration (Locle (Zhang et al., 2025), GNN-as-Judge) (Appendix D.2).

Evaluation and implementation. We report the mean and standard deviation of accuracy over three seeds; ground-truth labels of unlabeled nodes are used only for evaluation. FairRouter uses LLaMA3-8B (Grattafiori et al., 2024) as the default LLM, both experts stay frozen, and all configurations are selected on the validation set (Appendix D.3). In figures, the residual branch p^R is denoted as *Repair*.

Table 1: Node classification accuracy (%, mean±std over three seeds; Ave.: unweighted mean over the five datasets). Best results are **bolded** and second best underlined. Results of GCN, SGC, GLEM, LLaGA, and GraphGPT are from Xu & Ding (2026) (same splits).

Shot	Method	Cora	Citeseer	PubMed	Arxiv	Products	Ave.
3-shot	GCN	69.45±2.34	63.12±1.02	65.23±1.67	38.33±2.41	59.19±0.79	59.06
	SGC	67.21±1.25	63.07±0.95	65.34±1.18	39.16±1.49	57.42±1.12	58.44
	GLEM	67.81±0.92	53.09±1.39	63.85±1.02	36.37±1.96	52.46±1.93	54.72
	LLaGA	54.79±0.91	32.93±1.24	43.96±1.74	29.73±2.82	30.67±1.54	38.42
	GraphGPT	57.77±1.62	52.34±1.06	57.51±1.89	31.26±2.73	40.83±2.97	47.94
	LLM-GNN	73.37±1.76	66.80±0.65	73.37±0.68	47.40±1.48	54.93±0.45	63.17
	DMA	66.93±0.96	67.47±1.97	74.30±1.88	42.63±5.15	56.60±5.73	61.59
	Locle	74.30±1.90	69.30±0.37	72.97±1.92	51.70±0.98	63.80±1.63	66.41
	MoDis	75.50±4.74	62.67±1.45	72.97±5.35	25.43±2.08	25.87±1.11	52.49
	TA-GST	78.37±0.86	61.00±1.76	69.23±6.49	33.20±0.73	29.17±2.37	54.19
	GNN-as-Judge	74.37±1.19	64.07±1.82	88.93±1.30	53.80±3.82	**75.00±1.71**	71.23
	FairRouter	**79.37±0.17**	**73.37±0.12**	**90.80±1.04**	**58.13±0.12**	72.30±0.85	**74.79**
5-shot	GCN	73.69±1.03	63.24±0.87	70.58±0.49	45.67±0.41	68.26±1.54	64.29
	SGC	72.48±0.35	62.08±0.77	70.74±0.94	49.89±1.23	66.78±1.56	64.39
	GLEM	74.69±0.46	61.71±0.58	73.29±1.44	39.19±1.57	56.87±1.70	61.15
	LLaGA	62.88±2.19	43.71±4.36	58.63±1.05	33.74±2.45	37.29±3.12	47.25
	GraphGPT	60.17±1.44	51.83±2.24	57.39±3.67	36.25±1.87	44.78±2.34	50.08
	LLM-GNN	74.30±2.06	67.30±0.57	74.90±1.47	48.83±1.05	54.73±0.83	64.01
	DMA	67.93±1.39	67.80±2.38	74.57±2.38	47.20±3.02	61.40±3.35	63.78
	Locle	75.77±1.44	69.07±0.56	76.87±1.79	52.37±0.84	64.07±0.63	67.63
	MoDis	77.33±0.80	62.60±2.42	74.00±4.46	30.63±2.05	30.13±2.46	54.94
	TA-GST	77.80±1.72	63.23±1.65	68.77±9.18	34.27±3.25	34.03±1.58	55.62
	GNN-as-Judge	76.20±2.33	65.90±4.19	90.80±1.55	59.90±1.22	**75.10±0.95**	73.58
	FairRouter	**80.70±1.91**	**72.60±0.79**	**91.47±1.30**	**60.40±1.56**	72.43±0.41	**75.52**
10-shot	GCN	78.22±0.89	68.38±1.49	75.33±0.94	50.95±1.77	69.65±0.89	68.51
	SGC	78.49±0.37	67.44±0.60	74.98±1.91	51.89±1.23	67.91±0.48	68.14
	GLEM	78.11±0.73	66.83±0.61	74.17±2.39	47.73±1.09	60.22±1.89	65.41
	LLaGA	69.25±0.97	51.22±1.43	67.29±2.26	45.35±1.74	40.55±1.63	54.73
	GraphGPT	61.58±0.77	55.40±3.16	71.33±2.81	48.67±1.89	51.46±1.05	57.69
	LLM-GNN	76.47±1.50	68.33±0.95	75.90±1.28	52.23±1.82	55.13±0.39	65.61
	DMA	70.70±1.49	68.53±1.73	76.13±1.48	49.67±2.53	64.50±2.12	65.91
	Locle	78.30±1.30	68.87±0.52	76.40±1.69	56.23±1.13	65.37±1.14	69.03
	MoDis	79.53±0.58	65.23±3.86	76.23±2.87	38.67±0.84	37.40±2.35	59.41
	TA-GST	80.67±1.03	68.97±2.60	66.87±6.62	42.23±3.31	42.77±1.10	60.30
	GNN-as-Judge	79.17±1.23	68.17±1.01	91.43±1.10	58.80±5.40	**76.37±0.65**	74.79
	FairRouter	**81.27±1.36**	**73.47±1.25**	**91.63±0.83**	**61.60±0.86**	72.80±0.82	**76.15**

4.2 Main Results

Obs. 1: FairRouter achieves the best average accuracy under every label budget. As shown in Table 1, it ranks first on 12 of the 15 dataset–budget combinations and exceeds the strongest baseline, GNN-as-Judge, by 3.56/1.94/1.36 percentage points (pp) on average at 3/5/10 shots. It is also more stable (3-shot standard deviation $\leq$1.04, versus up to 3.82 for GNN-as-Judge), although on Arxiv at 5 shots and PubMed at 10 shots the margin lies within one standard deviation.

Obs. 2: The advantage is largest when labels are scarcest. With three labels per class, FairRouter already matches GNN-as-Judge with ten (74.79%), and at 3 shots it surpasses GNN-as-Judge by 5.00, 9.30, and 4.33 pp on Cora, Citeseer, and Arxiv, respectively.

Obs. 3: Pseudo-labels from a single fixed source are reliable only on some graphs. GNN self-training works well on small homophilous graphs (TA-GST is the runner-up on Cora at every budget) but collapses on the 40–47-class Arxiv and Products (25.43–42.77%), where it reinforces the GNN's initial errors (about 25–38% pseudo-label precision on Products; Figure 3). LLM-to-GNN methods (LLM-GNN and DMA) avoid this collapse but are capped by the quality of LLM annotations, trailing GNN-as-Judge by more than 14 pp on PubMed. Graph LLMs (LLaGA and GraphGPT) perform

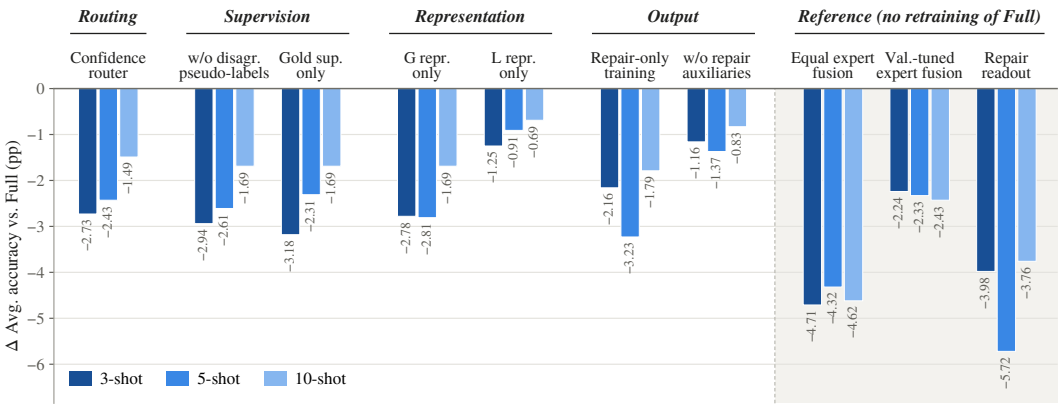


Figure 4: Core component ablations: change in average accuracy relative to the full model (pp); shaded variants are references without retraining (Conf.: confidence; disagr.: disagreement; sup.: supervision; repr.: representation; aux.: auxiliaries). Per-dataset results are in Appendix G.2.

worst, even below GCN, since aligning graph tokens with the LLM requires far more supervision than a few labels per class. Collaborative methods are the strongest baselines, confirming that the two modalities are complementary.

Obs. 4: The gains are largest where a fixed modality preference hurts most. FairRouter gains most over GNN-as-Judge on Citeseer (+9.30/+6.70/+5.30 pp) and Cora (+5.00/+4.50/+2.10 pp). On Citeseer, the frozen LLaMA3-8B alone reaches 72.77% (Table 2), above every baseline, yet GNN-as-Judge, which always follows the GNN on disagreement nodes, reaches only 64.07–68.17%, the failure mode of Figure 1(b). The exception is Products, where FairRouter trails GNN-as-Judge by 2.67–3.57 pp despite having the most precise pseudo-labels (92.0% versus about 67%; Figure 3). This gap stems from the learner rather than routing: GNN-as-Judge fine-tunes the 8B LLM to the fine-grained 47-category taxonomy (e.g., "Beauty" versus "All Beauty"), whereas FairRouter keeps both models frozen (Appendix K).

4.3 Pseudo-Label Quality

Overall precision. Figure 3 evaluates the pseudo-labels on which each method actually trains (3- and 10-shot results in Appendix E.1; branch-level statistics in Appendix E.2). FairRouter is the most precise on all five datasets (88.5–96.7%), and its gap over GNN-as-Judge grows with difficulty, from about 3 pp on PubMed to about 15 and 25 pp on Arxiv and Products. No fixed-source baseline is consistently second best: the LLM-driven DMA is close on Products but drops to about 65% on Cora, whereas the GNN-driven MoDis is close on Cora and PubMed but falls below 40% on Products and 50% on Arxiv. Which modality yields reliable pseudo-labels thus depends on the graph and the node, which is what motivates FairRouter; the low precision of MoDis and TA-GST also explains their collapse in Table 1.

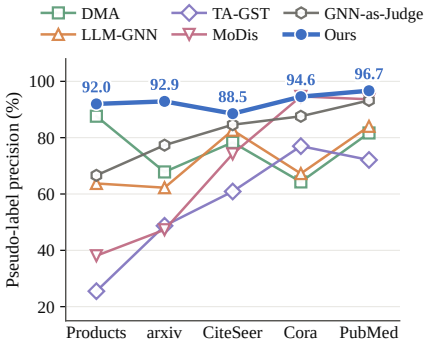


Figure 3: Precision of the pseudo-labels used by each method under the 5-shot setting.

Routing on disagreement nodes. At matched coverage on disagreement nodes (Appendix F), FairRouter is far more precise than following the more confident expert where confidence misleads (88% versus 45–52% on Cora), supporting Challenge 1. On Arxiv, where many disagreement nodes have no correct candidate, its trust score concentrates acceptance on the head of the ranking (91% versus 85%). On PubMed and Products, where the LLM dominates, plain confidence is 4–6 pp more precise, yet the full model remains more accurate than the confidence router (Section 4.4).

4.4 Ablation Study

Figure 4 reports the change in average accuracy on four benchmarks (settings in Appendix G.1); every retrained variant underperforms the full model at every budget.

Routing and supervision. A confidence router that accepts the *same number* of nodes costs 2.73/2.43/1.49 pp at 3/5/10 shots (up to 6.70 pp on Cora), so the gain comes from *which* nodes are

Table 2: Accuracy (%) of direct LLM inference (LLM) and FairRouter (+FR) under the 10-shot setting with different LLM backbones; Δ is the gain of FairRouter.

	Qwen3-8B			*Mistral-7B*			*LLaMA3-8B*		
Dataset	LLM	+FR	Δ	LLM	+FR	Δ	LLM	+FR	Δ
Cora	61.80±1.68	81.50±1.44	↑19.70	63.70±1.93	80.57±2.44	↑16.87	66.53±1.47	81.27±1.36	↑14.74
Citeseer	63.00±0.57	72.87±1.24	↑9.87	52.67±0.80	71.13±2.21	↑18.46	72.77±0.48	73.47±1.25	↑0.70
PubMed	85.70±1.10	89.80±0.86	↑4.10	83.50±0.57	90.50±0.73	↑7.00	84.47±1.25	91.63±0.83	↑7.16
Arxiv	59.30±0.54	62.80±0.70	↑3.50	25.23±0.91	54.33±0.82	↑29.10	51.73±0.58	61.60±0.86	↑9.87
Products	65.70±2.08	69.67±1.45	↑3.97	53.13±1.53	67.00±2.26	↑13.87	68.50±0.33	72.80±0.82	↑4.30
Avg.	67.10	**75.33**	↑8.23	55.65	**72.71**	↑17.06	68.80	**76.15**	↑7.35

selected and *which* modality is trusted, not from how many. Removing disagreement pseudo-labels is the most harmful single-component change under scarce labels (-2.94 pp at 3 shots), whereas additionally removing the agreement constraint changes the result by at most 0.30 pp: pseudo-supervision matters most on disagreement nodes, exactly where routing operates.

Representation and output. Removing the LLM representation (-2.78 pp at 3 shots; -4.10 pp on Arxiv) hurts more than removing the GNN representation (-1.25 pp), so the residual correction relies mainly on semantic cues. Training Residual as the sole output costs 2.16–3.23 pp, confirming the value of anchoring on the expert distributions.

Comparison with static fusion. A global mixture of the frozen models, even with validation-tuned weights, stays 2.24–2.43 pp below the full model, and this gap does *not* shrink with more labels: no global weighting can emulate node-adaptive decisions. Residual readout alone is weaker still (-3.76 to -5.72 pp), acting as a complementary correction rather than a standalone classifier (case F2 in Section 4.6).

4.5 Generalization and Robustness

LLM backbones. With Qwen3-8B (Yang et al., 2025) or Mistral-7B (Jiang et al., 2023) as the LLM expert, FairRouter improves over direct inference of its own LLM in 44 of the 45 backbone–dataset–budget settings (Table 2; Appendix I). The gain is largest for the weakest backbone (+17.06 pp for Mistral-7B at 10 shots), and the spread across backbones shrinks from 13.15 to 3.44 pp, as the router shifts trust toward the GNN when the LLM is weak.

Heterophilous graphs. On Cornell and Wisconsin, where neighborhood aggregation can mislead, we follow GNN-as-Judge and use H2GCN (Zhu et al., 2020) as the GNN expert (Figure 5). FairRouter improves over H2GCN by 12.2–19.2 pp under all budgets and outperforms the strong zero-shot LLM on Wisconsin under every budget (up to +10.8 pp), so structure still helps some nodes when routed selectively. On Cornell, it trails the zero-shot LLM by about 2 pp at 3 and 5 shots, as 15–25 labels are too few to identify the few nodes on which the GNN should be trusted, and surpasses it at 10 shots. FairRouter thus shifts trust toward the LLM when structure is unreliable, without any heterophily-specific design.

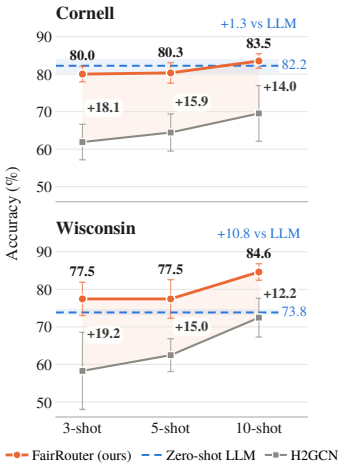


Figure 5: Accuracy on heterophilous graphs. Grey labels: gain over H2GCN; dashed line: zero-shot LLM.

Hyperparameter sensitivity. Figure 7 shows that (a) disabling disagreement supervision ($\rho_D = 0$) is the most harmful choice on every dataset (up to -5.1 pp), whereas any $\rho_D \geq 0.25$ stays within about 1 pp of the tuned result on Cora, PubMed, and Arxiv, as the per-source averaged loss tolerates moderate label noise; only Products, whose precision drops steeply, loses 1.1–1.8 pp for $\rho_D \geq 0.5$. The disagreement weight λ_D and its interaction with ρ_D are analyzed in Appendix H. (b) The auxiliary residual losses are indispensable on the 40-class Arxiv (-4.57 pp without them) but optional on the smaller graphs. (c) Accuracy is stable for learning rates in $[10^{-4}, 10^{-3}]$ and degrades at 10^{-2}. On Citeseer, several untuned settings even exceed the tuned one, suggesting that validation-based selection does not overfit the test set.

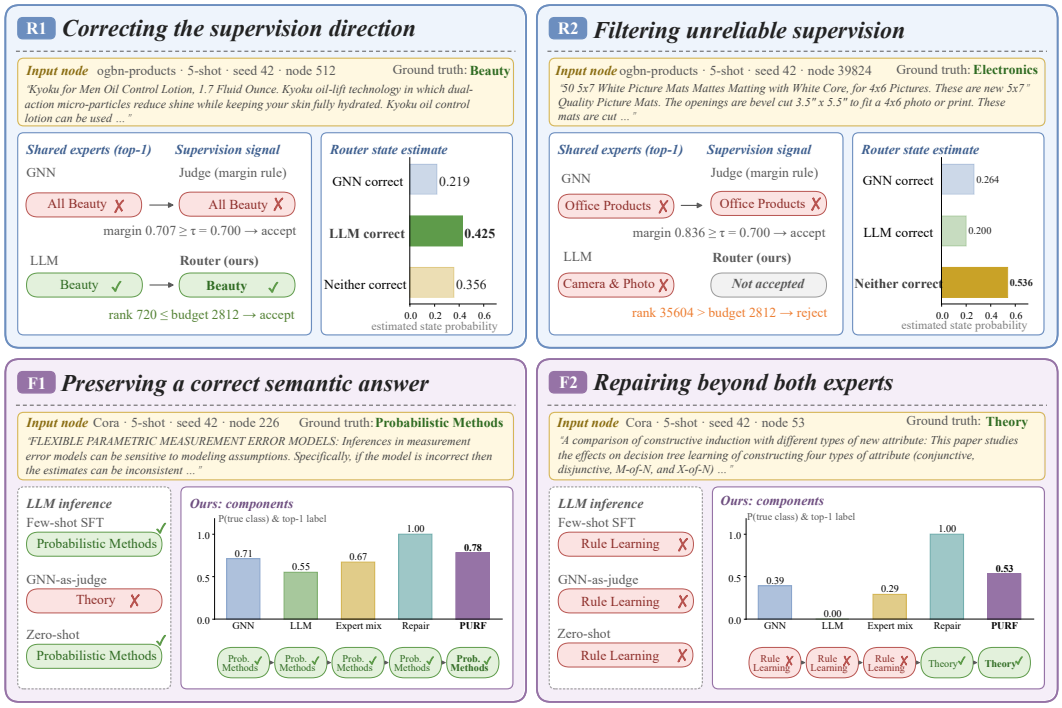


Figure 6: **Case studies. (R1, R2) Router supervision:** a GNN-preferring judge accepts the GNN label once its margin exceeds τ, whereas our router accepts a node only if its estimated correctness state ranks within the budget. **(F1, F2) Final prediction:** bars show $P(\text{true class})$ per component (Expert mix: p^{GL}; Residual: p^{R}; Final: p^{J}); chips show top-1 labels (✓/✗: correct/wrong).

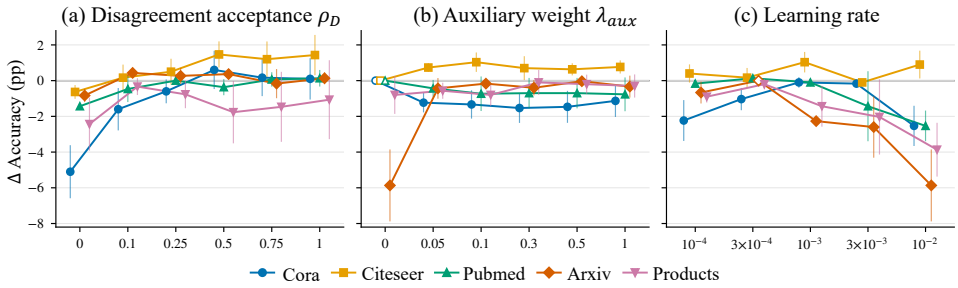


Figure 7: Hyperparameter sensitivity (5-shot): paired accuracy change (pp) w.r.t. the tuned configuration, averaged over three seeds (bars: ±1 SD; hollow markers: tuned setting).

4.6 Case Study

Figure 6 examines individual nodes (details in Appendix J). On Products, the router redirects supervision from a wrong GNN label (*All Beauty*) to the correct LLM label (*Beauty*) (R1), and rejects a node on which both models are wrong by assigning the highest probability to the *neither-correct* state (R2), whereas a GNN-preferring judge accepts both wrong labels. On Cora, FairRouter keeps a correct answer that the LLM retrained by GNN-as-Judge loses (F1), and its residual branch recovers a node that both models misclassify (F2; Challenge 2).

5 Conclusion

We presented FairRouter, which replaces the fixed-teacher assumption of GNN–LLM pseudo-labeling on text-attributed graphs with node-wise fair routing: a learnable router decides for each node whether to trust a pseudo-label and which modality to follow, and a multimodal residual classifier learns from the routed supervision while correcting errors shared by both modalities. Across five benchmarks, FairRouter achieves the best average accuracy under all label budgets, with the largest gains when labels are scarcest, and the most precise pseudo-labels on every dataset. Future work includes using routed pseudo-labels for LLM fine-tuning and more-than-two model routing.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript for grammar and readability and to assist with polishing the language. We also used AI coding assistants for plotting scripts; all such code was reviewed and tested by the authors. We did not use generative AI tools to generate research ideas or hypotheses, design the methodology or experiments, implement the method, generate or reformat datasets, or interpret results; generating synthetic data, formulating mathematical claims, and writing proofs are not applicable to this work. The LLMs evaluated in this paper (LLaMA3-8B, Qwen3-8B, and Mistral-7B) are components of the studied method and baselines rather than research or writing aids. We have reviewed all AI-assisted content and verified every reported number against our experimental records, and we take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies few-shot node classification on publicly available benchmark graphs (citation networks, a product co-purchasing network, and webpage graphs) that are widely used in prior research. It involves no human subjects, private data, or personally identifiable information. Since FairRouter derives pseudo-labels from pretrained LLMs, biases present in these models may propagate into the pseudo-labels. The trust scores of our router provide an explicit mechanism to reject unreliable predictions, but they do not remove such biases. We are not aware of other ethical concerns raised by this work.

REPRODUCIBILITY STATEMENT

The complete source code is available at the anonymous repository linked in the abstract.

REFERENCES

Jonathan Chang and David Blei. Relational topic models for document networks. In *Artificial intelligence and statistics*, pp. 81–88, 2009.

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235, pp. 7809–7823, 2024a.

Zhikai Chen, Haitao Mao, Hongzhi Wen, Haoyu Han, Wei Jin, Haiyang Zhang, Hui Liu, and Jiliang Tang. Label-free node classification on graphs with large language models (llms). In *International Conference on Learning Representations*, volume 2024, pp. 31632–31655, 2024b.

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

Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections. In *International Conference on Learning Representations*, volume 2025, pp. 26186–26203, 2025.

Baoquan Gong, Xiyuan Gao, Pengfei Zhu, Qinghua Hu, and Bing Cao. Multimodal negative learning. *Advances in Neural Information Processing Systems*, 38:153226–153257, 2026.

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.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330, 2017.

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

Zhihui Hu, Yao Fu, Hong Zhao, Xiaoyu Cai, Weihao Jiang, and Shiliang Pu. Liberate pseudo labels from over-dependence: Label information migration on sparsely labeled graphs. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 833–842, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Qing-Yuan Jiang, Longfei Huang, and Yang Yang. Rethinking multimodal learning from the perspective of mitigating classification ability disproportion. *Advances in Neural Information Processing Systems*, 38:129580–129607, 2026.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

Quan Li, Lingwei Chen, Shixiong Jing, and Dinghao Wu. Pseudo-labeling with graph active learning for few-shot node classification. In *2023 IEEE International Conference on Data Mining (ICDM)*, pp. 1115–1120. IEEE, 2023a.

Quan Li, Tianxiang Zhao, Lingwei Chen, Junjie Xu, and Suhang Wang. Enhancing graph neural networks with limited labeled data by actively distilling knowledge from large language models. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 741–746. IEEE, 2024.

Yayong Li, Jie Yin, and Ling Chen. Informative pseudo-labeling for graph neural networks with few labels. *Data Mining and Knowledge Discovery*, 37(1):228–254, 2023b.

Gen Liu, Zhongying Zhao, Hui Zhou, Chao Li, and Qingtian Zeng. Can pseudo-label be more reliable? a simple yet effective topology-aware graph self-training method. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 23685–23693, 2026a.

Hongrui Liu, Binbin Hu, Xiao Wang, Chuan Shi, Zhiqiang Zhang, and Jun Zhou. Confidence may cheat: Self-training on graph neural networks under distribution shift. In *Proceedings of the ACM web conference 2022*, pp. 1248–1258, 2022.

Ruyue Liu, Rong Yin, Xiangzhen Bo, Xiaoshuai Hao, Yong Liu, Jinwen Zhong, Can Ma, and Weiping Wang. Sstag: Structure-aware self-supervised learning method for text-attributed graphs. *Advances in Neural Information Processing Systems*, 38:13883–13906, 2026b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Donald Loveland, Yao-An Yang, and Danai Koutra. Glance for context: Learning when to leverage llms for node-aware gnn-llm fusion. In *International Conference on Learning Representations*, volume 2026, pp. 90402–90428, 2026.

Yihong Luo, Yuhan Chen, Siya Qiu, Yiwei Wang, Chen Zhang, Yan Zhou, Xiaochun Cao, and Jing Tang. Fast graph sharpness-aware minimization for enhancing and accelerating few-shot node classification. *Advances in Neural Information Processing Systems*, 37:132364–132387, 2024.

Haitao Mao, Zhikai Chen, Wei Jin, Haoyu Han, Yao Ma, Tong Zhao, Neil Shah, and Jiliang Tang. Demystifying structural disparity in graph neural networks: Can one size fit all? *Advances in neural information processing systems*, 36:37013–37067, 2023.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. In *International Conference on Learning Representations*, 2020.

Hongbin Pei, Yuheng Xiong, Pinghui Wang, Jing Tao, Jialun Liu, Huiqi Deng, Jie Ma, and Xiaohong Guan. Memory disagreement: A pseudo-labeling measure from training dynamics for semi-supervised graph learning. In *Proceedings of the ACM web conference 2024*, pp. 434–445, 2024.

Simone Ricci, Niccolò Biondi, Federico Pernici, and Alberto Del Bimbo. Mitigating negative flips via margin preserving training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8721–8730, 2026.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S. Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

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.

Qian Shao, Qiyuan Chen, Jiahe Chen, Zepeng Li, Qianqian Tang, Hongxia Xu, and Jian Wu. Cola: Co-calibrated logit adjustment for long-tailed semi-supervised learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zeang Sheng, Weiyang Guo, Yingxia Shao, Wentao Zhang, and Bin Cui. Llms are noisy oracles! llm-based noise-aware graph active learning for node classification. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2526–2537, 2025.

Yuan Sui and Bryan Hooi. Conversation for non-verifiable learning: Self-evolving large language models through meta-evaluation. In *Forty-third International Conference on Machine Learning*, 2026.

Gu Tang, Jinghe Wang, Xiaoying Gan, Bin Lu, Ze Zhao, Luoyi Fu, Xinbing Wang, and Chenghu Zhou. R2mr: Review and rewrite modality for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 1337–1348, 2025.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pp. 491–500, 2024.

Fali Wang, Tianxiang Zhao, and Suhang Wang. Distribution consistency based self-training for graph neural networks with sparse labels. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 712–720, 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*, 2025.

Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. *Advances in Neural Information Processing Systems*, 34:23768–23779, 2021.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 13723–13732. IEEE, 2020.

Xing Wei, Chunchun Chen, Rui Fan, Xiaofeng Cao, Sourav Medya, and Wei Ye. Preference-driven knowledge distillation for few-shot node classification. *Advances in Neural Information Processing Systems*, 38:103297–103327, 2026.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pp. 6861–6871, 2019.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, volume 2024, pp. 23650–23678, 2024.

Ruiyao Xu and Kaize Ding. Gnn-as-judge: Unleashing the power of llms for graph learning with gnn feedback. *arXiv preprint arXiv:2604.08553*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. Network representation learning with rich text information. In *Ijcai*, volume 2015, pp. 2111–2117, 2015.

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

Jianxiang Yu, Yuxiang Ren, Chenghua Gong, Jiaqi Tan, Xiang Li, and Xuecang Zhang. Leveraging large language models for node generation in few-shot learning on text-attributed graphs. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 13087–13095, 2025.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Qinghua Hu, Huazhu Fu, Joey Tianyi Zhou, and Xi Peng. Provable dynamic fusion for low-quality multimodal data. In *International conference on machine learning*, pp. 41753–41769, 2023.

Taiyan Zhang, Renchi Yang, Yurui Lai, Mingyu Yan, Xiaochun Ye, and Dongrui Fan. Leveraging large language models for effective label-free node classification in text-attributed graphs. In *Proceedings of the 48th international ACM SIGIR conference on research and development in information retrieval*, pp. 698–708, 2025.

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. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*, 2023.

Ze Zhao, Bin Lu, Xiaoying Gan, Gu Tang, Luoyi Fu, and Xinbing Wang. Chainsformer: Numerical reasoning on knowledge graphs from a chain perspective. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 2963–2976. IEEE, 2025.

Jiong Zhu, Yujun Yan, Lingxiao Zhao, Mark Heimann, Leman Akoglu, and Danai Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in neural information processing systems*, 33:7793–7804, 2020.

Appendix Contents

A Feature Constructions for FairRouter **15**

A.1 Notation and Feature Organization 15

A.2 Probability and Distribution Extractor 15

A.3 Neighborhood and Structural Extractor 16

A.4 Gold Prototype Extractor 17

A.5 Assembly, Feature Modes, and Representation Inputs 18

B Implementation Details for FairRouter **19**

B.1 Agreement Quality Scorer Configurations 19

B.2 Disagreement Scorers and Score Transformations 21

B.3 Preference Routing and Selective Acceptance 24

B.4 Multi-View Augmentation 25

C Implementation Details for Residual Classifier **27**

C.1 Multi-modal Residual Classifier 27

C.2 Configurations and Details 28

D Experimental Details **28**

D.1 Datasets and Splits 28

D.2 Baselines 29

D.3 Implementation Details 30

D.4 Pseudo-Label Evaluation Protocol 30

E Pseudo-Label Quality Analysis **31**

E.1 Pseudo-Label Precision under Other Label Budgets 31

E.2 Branch-Level Pseudo-Label Statistics 31

F Precision–Coverage Analysis on Disagreement Nodes **33**

G Ablation Study Details **34**

G.1 Ablation Settings 34

G.2 Full Ablation Results 34

H Additional Hyperparameter Sensitivity **37**

I Generalization across LLM Backbones **38**

J Case Studies **38**

K Limitations and Future Work **39**

A FEATURE CONSTRUCTIONS FOR FAIRROUTER

This appendix details the three evidence extractors of FairRouter. All of them are deterministic and have no trainable parameters.

A.1 NOTATION AND FEATURE ORGANIZATION

Let $C \geq 2$ be the number of classes and $E \in \{G, L\}$ denote the graph or language modality. For node i, write $p_E(c) = p_i^E(c)$ and let $p_{E,(1)} \geq p_{E,(2)}$ denote the two largest probabilities in its complete class distribution. The candidate prediction c is either $\hat{y}_i^G$ or $\hat{y}_i^L$. All logarithms are natural, and $\ell_\epsilon(x) = \log \max(x, \epsilon)$ with $\epsilon = 10^{-12}$ provides numerical protection. We use $[\,\cdot\,;\,\cdot\,]$ for concatenation and $\mathbb{I}[\cdot]$ for an indicator.

Consistent with the main text, $\phi(i, c)$ abbreviates $\phi(i, c \mid p_i^G, p_i^L, \mathcal{C}_i)$ and $\psi(i)$ abbreviates $\psi(i \mid p_i^G, p_i^L, \mathcal{C}_i)$. The context $\mathcal{C}_i$ supplies the graph connectivity, neighboring predictions, node representations, and labeled few-shot references required by the neighborhood and prototype extractors. The three extractors return candidate blocks and shared descriptors:

$$\begin{aligned} \phi(i, c) &= [\phi_{\text{prob}}(i, c); \phi_{\text{nbr}}(i, c); \phi_{\text{gold}}(i, c)], \\ \psi(i) &= [\psi_{\text{prob}}(i); \psi_{\text{nbr}}(i); \psi_{\text{gold}}(i)]. \end{aligned} \tag{7}$$

Their dimensions are summarized in Table 3. The split into candidate-specific and shared parts does not follow the functional grouping below: the six individual-confidence statistics are node-level quantities but are included in every candidate block.

Extractor	$\dim \phi_k$	$\dim \psi_k$	Agreement	Disagreement
Probability	16	9	25	41
Neighborhood	3	3	6	9
Prototypes	8	4	12	20
Total	27	16	43	70

Table 3: Dimensions of the evidence vectors. Agreement nodes use one candidate block; disagreement nodes use two.

A.2 PROBABILITY AND DISTRIBUTION EXTRACTOR

The probability extractor $\mathcal{F}_{\text{prob}}$ serves three functions: (i) *individual confidence*, describing how concentrated and confident each modality's prediction is; (ii) *inter-modality discrepancy*, measuring differences between the full predictive distributions and confidence levels; and (iii) *candidate-specific support*, characterizing how strongly each model supports each candidate label. These features capture the absolute reliability, relative competence, and complementary evidence from the two modalities at each node.

Individual confidence: six statistics. For each modality, we compute normalized entropy, the top-two probability margin, and the maximum probability:

$$\begin{aligned} H_E(i) &= -\frac{1}{\log C} \sum_{k=1}^{C} p_E(k) \ell_\epsilon(p_E(k)), \\ M_E(i) &= p_{E,(1)} - p_{E,(2)}, \qquad U_E(i) = p_{E,(1)}. \end{aligned} \tag{8}$$

These describe the dispersion, decisiveness, and peak confidence of each modality's prediction. The six statistics $[H_G, M_G, U_G, H_L, M_L, U_L]$ are included in each candidate block, so the disagreement vector contains them twice.

Inter-modality discrepancy: nine statistics. Let $m(k) = \frac{1}{2}(p_G(k) + p_L(k))$. We compute

$$\begin{aligned} J(i) &= \frac{1}{2} \sum_{E \in \{G,L\}} \sum_{k=1}^{C} p_E(k) \big[\ell_\epsilon(p_E(k)) - \ell_\epsilon(m(k))\big], \\ B(i) &= \sum_{k=1}^{C} \sqrt{p_G(k) p_L(k)}, \\ D_1(i) &= \sum_{k=1}^{C} |p_G(k) - p_L(k)|, \\ D_2(i) &= \sqrt{\sum_{k=1}^{C} (p_G(k) - p_L(k))^2}. \end{aligned} \tag{9}$$

Here J is Jensen–Shannon divergence, B is the Bhattacharyya overlap coefficient, and D_1, D_2 are distribution distances. We use J without the division by $\log 2$, B without the negative-log transform, and D_1 without the factor $1/2$. The full shared descriptor is

$$\begin{aligned} \psi_{\text{prob}}(i) = \big[&J, B, D_1, D_2, \min(U_G, U_L), \max(U_G, U_L), \\ &|U_G - U_L|, H_G + H_L, |H_G - H_L|\big]. \end{aligned} \tag{10}$$

The minimum and maximum are taken over each model's own top-1 probability, even when the two models predict different classes.

Candidate-specific support: ten statistics. For each modality E and candidate label c, we compute five quantities:

$$\begin{aligned} P_E(i,c) &= p_E(c), \qquad L_E(i,c) = \ell_\epsilon(p_E(c)), \\ R_E(i,c) &= \frac{1}{C-1} \sum_{k=1}^{C} \mathbb{I}[p_E(k) > p_E(c)], \\ A_E(i,c) &= p_{E,(1)} - p_E(c), \\ T_E(i,c) &= p_E(c) - p_{E,(2)}. \end{aligned} \tag{11}$$

R_E is a normalized rank, with zero indicating that no class has a strictly greater probability. Equal probabilities share the same rank. A_E measures the shortfall of candidate c relative to the top-1 class of modality E; T_E measures its advantage over the second-largest probability in the full distribution. Here $p_{E,(2)}$ is always taken from the full distribution, including when c is the top-1 class.

The same five quantities are evaluated under both modalities, so each candidate is described by its support under both the GNN and the LLM. The block is ordered as

$$\begin{aligned} \phi_{\text{prob}}(i,c) = \big[&P_G, L_G, P_L, L_L, R_G, R_L, A_G, A_L, \\ &T_G, T_L, H_G, M_G, U_G, H_L, M_L, U_L\big]. \end{aligned} \tag{12}$$

Thus this block contains ten modality-conditioned values and six individual-confidence values. Pairwise differences are not included here; the `compact` feature mode adds selected directional differences (Appendix A.5).

A.3 Neighborhood and Structural Extractor

The neighborhood extractor $\mathcal{F}_{\text{nbr}}$ uses *node degree*, an *isolation indicator*, and *neighborhood prediction dispersion*. These features evaluate whether a candidate prediction is supported by the local structural context. They are computed from the GNN's predictions on neighboring nodes and require no neighbor labels.

Edges are symmetrized and deduplicated, and self-loops are kept. Let $\mathcal{N}(i)$ be the resulting neighbor set and $d_i = |\mathcal{N}(i)|$. Define the neighbor prediction distribution as

$$\nu_i(k) = \begin{cases} \dfrac{1}{d_i} \displaystyle\sum_{u \in \mathcal{N}(i)} \mathbb{I}[\hat{y}_u^G = k], & d_i > 0, \\ 0, & d_i = 0. \end{cases} \tag{13}$$

The candidate support $V(i,c) = \nu_i(c)$ is the fraction of neighbors whose GNN prediction equals c; both candidates of a disagreement node are evaluated against the same distribution. The candidate block $\phi_{\mathrm{nbr}}(i,c)$ consists of $V(i,c)$ padded with two constant entries, which carry no information.

The shared structural statistics are

$$H_{\mathcal{N}}(i) = \begin{cases} -\dfrac{1}{\log C}\displaystyle\sum_{k=1}^{C} \nu_i(k)\ell_\epsilon(\nu_i(k)), & d_i > 0, \\ 1, & d_i = 0, \end{cases}$$
$$\psi_{\mathrm{nbr}}(i) = [H_{\mathcal{N}}(i), \log(1+d_i), \mathbb{I}[d_i = 0]]. \tag{14}$$

$H_{\mathcal{N}}$ is the entropy of the neighbors' hard predictions, not the average entropy of their probability vectors. Isolated nodes receive zero votes, unit entropy, and an isolation flag.

A.4 Gold Prototype Extractor

The prototype extractor $\mathcal{F}_{\mathrm{gold}}$ assesses candidate reliability by comparing the node with few-shot gold references in the frozen representation space, capturing prototype proximity, instance-level similarity, and global class separation. These features provide a labeled reference for judging candidate predictions.

It operates on the self-supervised GNN representations. Let z_u^0 denote the representation of a labeled support node computed on the clean graph, and $\mathcal{S}_c = \{u \in \mathcal{S} : y_u = c\}$. For a nonempty class, its prototype is

$$\mu_c = \frac{1}{|\mathcal{S}_c|}\sum_{u \in \mathcal{S}_c} z_u^0. \tag{15}$$

which is normalized after averaging. For any vector z, define $\bar{z} = z/\max(\|z\|_2, \epsilon)$. The representation z_i of the evaluated node is computed on the perturbed graph for graph-perturbed views and on the clean graph otherwise.

Candidate-class prototype evidence: eight statistics. Define the candidate-class prototype distance and similarity by

$$D(i,c) = \begin{cases} 1 - \bar{z}_i^\top \bar{\mu}_c, & |\mathcal{S}_c| > 0, \\ 2, & |\mathcal{S}_c| = 0, \end{cases}$$
$$S(i,c) = \begin{cases} 1 - D(i,c), & |\mathcal{S}_c| > 0, \\ 0, & |\mathcal{S}_c| = 0. \end{cases} \tag{16}$$

The nearest same-class and other-class support distances are

$$D_{\mathrm{same}}(i,c) = \min_{u \in \mathcal{S}_c}(1 - \bar{z}_i^\top \bar{z}_u^0),$$
$$D_{\mathrm{other}}(i,c) = \min_{u \in \mathcal{S} \setminus \mathcal{S}_c}(1 - \bar{z}_i^\top \bar{z}_u^0), \tag{17}$$

where a minimum over an empty reference set is assigned the sentinel value 2. The normalized prototype rank is

$$R_{\mathrm{gold}}(i,c) = \frac{1}{C-1}\sum_{k=1}^{C} \mathbb{I}[D(i,k) < D(i,c)]. \tag{18}$$

Tied distances share the same rank, and classes without support nodes enter through their sentinel value. The candidate block is

$$\phi_{\mathrm{gold}}(i,c) = \big[S(i,c), D(i,c), D_{\mathrm{same}}(i,c), D_{\mathrm{other}}(i,c), R_{\mathrm{gold}}(i,c), m_c^{\mathrm{proto}}, m_c^{\mathrm{same}}, m_c^{\mathrm{other}}\big], \tag{19}$$

with $m_c^{\mathrm{proto}} = m_c^{\mathrm{same}} = \mathbb{I}[|\mathcal{S}_c| = 0]$ and $m_c^{\mathrm{other}} = \mathbb{I}[|\mathcal{S} \setminus \mathcal{S}_c| = 0]$ indicating missing references.

Shared reference geometry: four statistics. Let $D_{(1)}(i) \le D_{(2)}(i)$ be the two smallest distances among all class prototypes, including the missing-class sentinels. Let

$$D_{\text{support}}(i) = \min_{u \in \mathcal{S}} (1 - \bar{z}_i^\top \bar{z}_u^0), \tag{20}$$

again taking value 2 for an empty support set. Then

$$\psi_{\text{gold}}(i) = [D_{(1)}(i), D_{(2)}(i) - D_{(1)}(i), D_{\text{support}}(i), \mathbb{I}[|\mathcal{S}| = 0]]. \tag{21}$$

These describe proximity to known classes, separation between the closest classes, and proximity to any labeled reference node.

Reference construction. Class membership of the references is determined by gold labels, not by candidate predictions. Prototypes and support representations are always computed on the clean graph and shared by all views: text-perturbed views keep the clean graph representation of the evaluated node, and graph-perturbed views compare its perturbed representation against the clean references. When the features of a support node itself are computed for router training, the node is not excluded from its class prototype or from the nearest-support search, and the prototypes are not recomputed within the calibration folds of Appendix B.2.

A.5 Assembly, Feature Modes, and Representation Inputs

The complete evidence vector is

$$\mathbf{x}_i^{\text{tab}} = \begin{cases} [\phi(i, \hat{y}_i^G); \psi(i)], & \hat{y}_i^G = \hat{y}_i^L, \\ [\phi(i, \hat{y}_i^G); \phi(i, \hat{y}_i^L); \psi(i)], & \hat{y}_i^G \neq \hat{y}_i^L. \end{cases} \tag{22}$$

Its dimensions are $27 + 16 = 43$ and $2 \times 27 + 16 = 70$. Because some entries are duplicated or constant, the number of informative features is smaller.

Feature modes. Each scorer uses one of the following feature modes, to which its fitted transformation (described below) is then applied.

Mode	Agreement	Disagreement
`prob`	25	41
`full`	43	70
`compact`	15	24
`context`	18	28
`embed`	39	48

The `prob` mode retains the probability extractor's modality and shared blocks; `full` uses all entries. The `compact` mode selects probability evidence and adds interactions. For disagreement, it appends the four directional contrasts

$$\begin{aligned} \delta_1 &= L_G(i, \hat{y}_i^G) - L_G(i, \hat{y}_i^L), \quad \delta_2 = L_L(i, \hat{y}_i^G) - L_L(i, \hat{y}_i^L), \\ \delta_3 &= P_G(i, \hat{y}_i^G) - P_G(i, \hat{y}_i^L), \quad \delta_4 = P_L(i, \hat{y}_i^G) - P_L(i, \hat{y}_i^L), \end{aligned} \tag{23}$$

plus $\delta_1(1 - H_G)$ and $\delta_2(1 - H_L)$. These compare the two candidate labels within each modality and complement the paired, cross-modality support recorded for each modality. For agreement, the additional interactions are $P_G(i, c)P_L(i, c)$, $\min(P_G(i, c), P_L(i, c))$, and $|H_G - H_L|$. The `context` mode adds neighbor votes, neighborhood entropy, and log degree to the compact evidence.

Representation inputs. Besides the explicit evidence, a scorer may use the frozen graph representation $\mathbf{h}_i^G \in \mathbb{R}^{128}$ and language representation $\mathbf{h}_i^L \in \mathbb{R}^{4096}$; the concatenation in the main text lists all available inputs. Scorers in the `embed` mode append the two fixed projections below, the Citeseer trust scorer uses the learned language projection described next, and all other scorers use the explicit evidence only. The 43/70-dimensional counts exclude these representation inputs.

Fixed projections for the embed mode. The `embed` mode appends fixed signed sparse projections of the graph and language representations to the compact features. For an input $\mathbf{h} \in \mathbb{R}^D$, a fixed bucket assignment $b_j \in \{1, \dots, m\}$ and sign $s_j \in \{-1, 1\}$ define

$$u_k = \sum_{j:b_j=k} s_j h_j, \qquad \Pi(\mathbf{h}) = \frac{\mathbf{u}}{\max(\|\mathbf{u}\|_2, 10^{-8})}. \tag{24}$$

Each input coordinate is assigned to one output bucket with a fixed sign. The graph and language projections use $m = 8$ and $m = 16$, with seeds 171 and 173, respectively. The projection parameters are not learned. Concatenating the resulting 24 representation features with the compact features yields 39 raw dimensions for agreement and 48 for disagreement. The scorer's fitted feature transformation is subsequently applied to this concatenated vector, so its final dimension can be smaller.

Learned language projection for the Citeseer trust scorer. The Citeseer trust scorer maps the language representation through a learned bottleneck:

$$\Pi^L(\mathbf{h}_i^L) = \mathrm{GELU}\big(W_L\,\mathrm{LayerNorm}(\mathbf{h}_i^L) + \mathbf{b}_L\big), \qquad W_L \in \mathbb{R}^{16\times 4096}, \quad \mathbf{b}_L \in \mathbb{R}^{16}. \tag{25}$$

Its 16-dimensional output is concatenated with the 70 preprocessed explicit features, yielding an 86-dimensional input to the fusion network. The explicit features are standardized with statistics from the source datasets, which are kept fixed during target adaptation, and clipped to $[-20, 20]$. The projection and fusion parameters are trained with the trust loss on labeled observations, while the GNN and LLM remain frozen. No graph representation is used by this scorer.

Fitted transformations. The transformations are fitted on the labeled (support) observations: they estimate weighted moments, remove near-constant columns, and greedily discard columns whose absolute weighted standardized correlation with a retained column reaches 0.995. Standardized values are clipped to $[-8, 8]$. For configurations using a rank transform, retained features are mapped using their sorted training reference values $\mathcal{R}_j$:

$$T_j(x) = \frac{\#\{r \in \mathcal{R}_j : r < x\} + \#\{r \in \mathcal{R}_j : r \le x\}}{|\mathcal{R}_j|} - 1. \tag{26}$$

Consequently, fitted input dimensions can be smaller than the raw dimensions above. The Citeseer neural trust model clips source-standardized explicit features to $[-20, 20]$; the Arxiv agreement model clips standardized explicit features to $[-12, 12]$; the Products logistic trust model uses standardized probability features clipped to $[-8, 8]$. No unlabeled or test nodes are used to fit these transformations.

B Implementation Details for FairRouter

B.1 Agreement Quality Scorer Configurations

Task and supervision. To distinguish correct agreements from shared errors, agreement scoring evaluates the correctness of the models' common prediction. For an agreement observation i, let $c_i = \hat{y}_i^G = \hat{y}_i^L$. Its binary target is

$$y_A(i) = \mathbb{I}[c_i = y_i]. \tag{27}$$

The final scoring function is dataset-specific: Cora and Citeseer use fixed confidence rules, while PubMed, Arxiv, and Products use learned binary classifiers. All configurations produce a scalar score $s_A(i)$ for ranking agreement nodes.

Configuration overview. Table 4 summarizes the final agreement scorers. Source training uses the other four datasets, excluding the target dataset; target training uses only the target dataset. Input dimensions refer to the features actually supplied to the classifier after the selected transformation.

Fixed confidence rules. The Cora scorer combines confidence from both modalities in the common prediction through their geometric mean:

$$s_A(i) = \sqrt{p_i^G(c_i)\, p_i^L(c_i)}. \tag{28}$$

Dataset	Scorer	Training	Input mode	Dimension
Cora	Fixed rule	None	Common-label probabilities	2
Citeseer	Fixed rule	None	Language probability	1
PubMed	Logistic regression	Source	`embed`	37
Arxiv	Extra Trees	Target	Full tabular	43
Products	Extra Trees	Source	`compact`	13

Table 4: Final agreement configurations. Dimensions for fixed rules count their scalar probability inputs. The PubMed and Products transformations reduce their raw feature dimensions from 39 to 37 and from 15 to 13, respectively.

The Citeseer scorer uses the language model's confidence:

$$s_A(i) = p_i^L(c_i). \tag{29}$$

These rules require no classifier fitting or representation projection. Their scores enter the same selective acceptance procedure as those of the learned classifiers.

PubMed: logistic regression with projected representations. The PubMed scorer uses the `embed` feature mode. Its raw input concatenates 15 compact explicit features with the fixed graph and language projections of dimensions 8 and 16, respectively. The resulting 39-dimensional vector undergoes the fitted feature selection and rank transformation described in Section A.5, yielding 37 retained features.

Let $\boldsymbol{\xi}_i$ denote the transformed input. The classifier is an L2-regularized logistic regression model with inverse regularization strength $C_{\mathrm{LR}} = 0.01$, a maximum of 1500 optimization iterations, and random seed 42. It is trained on agreement observations from Cora, Citeseer, Arxiv, and Products. The final ranking score is its decision function:

$$s_A(i) = \mathbf{w}_A^\top \boldsymbol{\xi}_i + b_A. \tag{30}$$

This logit is used directly for ranking, without a sigmoid or calibration.

Arxiv: shallow Extra Trees on evidence. The Arxiv scorer uses all 43 agreement explicit features and no additional representation branch. Feature means and standard deviations are estimated on the labeled Arxiv observations, and standardized values are clipped to $[-12, 12]$.

The classifier is an Extra Trees ensemble with 160 trees, maximum depth 2, minimum leaf size 5, and random seed 42. The feature-subsampling setting is `max_features="sqrt"`. Training uses agreement observations from Arxiv only. Let $\pi_A(i)$ be the ensemble's predicted probability of $y_A(i) = 1$. The final score is

$$s_A(i) = \pi_A(i). \tag{31}$$

Products: Extra Trees on compact evidence. The Products scorer uses the `compact` feature mode, without graph or language representation projections. Its 15 raw features undergo gold-fitted removal of near-constant and highly correlated columns, leaving 13 features. The retained features are standardized and clipped to $[-8, 8]$; no rank transformation is used.

The classifier is an Extra Trees ensemble with 200 trees, maximum depth 8, minimum leaf size 3, and random seed 42. All input features are available when considering a split, corresponding to `max_features=1.0`. Training uses agreement observations from Cora, Citeseer, PubMed, and Arxiv. The final score is the clipped log-odds of the predicted positive-class probability:

$$\begin{aligned} \bar{\pi}_A(i) &= \min(1 - \epsilon_s, \max(\epsilon_s, \pi_A(i))), \\ s_A(i) &= \log \frac{\bar{\pi}_A(i)}{1 - \bar{\pi}_A(i)}, \qquad \epsilon_s = 10^{-6}. \end{aligned} \tag{32}$$

This transformation preserves the ordering of probabilities within the clipping interval and assigns equal scores to values clipped to the same endpoint.

Training observations and weights. Learned agreement scorers use eligible gold observations from the clean and augmented views. An observation is eligible only when its environment-specific candidate predictions agree; its target is recomputed using the unchanged gold label. Consequently, different views of the same node may enter different routing branches.

Let $\mathcal{S}_{d,A}$ denote the eligible agreement observations from training dataset d, and let $\mathcal{D}_A$ denote the datasets used by the selected scorer. Each source dataset receives equal total training mass for PubMed and Products: $\alpha_{d,A} = 1/4$. For Arxiv, the target dataset receives mass $\alpha_{d,A} = 1$. Within each dataset, clean and augmented observations have equal weight. The sample weights supplied to the estimators are

$$\widetilde{w}_{i,A} = N_A \frac{\alpha_{d,A}}{|\mathcal{S}_{d,A}|}, \qquad N_A = \sum_{d \in \mathcal{D}_A} |\mathcal{S}_{d,A}|, \qquad i \in \mathcal{S}_{d,A}. \tag{33}$$

Thus the weights sum to N_A, while preserving the intended relative mass of each dataset. The logistic classifier uses weighted fitting with L2 regularization; the tree ensembles use the supplied sample weights during fitting. Classifier parameters and fitted input transformations use gold training observations only.

Ranking and selective acceptance. The agreement branch uses its scorer output directly, without the affine calibration applied to the disagreement heads. The final scores therefore have different numerical interpretations across datasets: fixed confidence scores, logistic decision values, ensemble probabilities, or transformed ensemble probabilities. They are used only to rank nodes within the corresponding dataset's agreement branch and need not be calibrated probabilities.

Let $\mathcal{U}_A$ be the agreement population. The accepted agreement set is

$$\mathcal{P}_A = \mathrm{TopK}(\mathcal{U}_A, s_A, \lceil \rho_A |\mathcal{U}_A| \rceil), \tag{34}$$

with score ties resolved by node ID. The acceptance ratio ρ_A is selected separately for every dataset, label budget, and seed; the selected values are listed in Table 7. These ratios are selected from a finite grid on validation data, prioritizing pseudo-label precision, followed by the Wilson lower bound and coverage, subject to at least 20 accepted validation observations. Each accepted node receives the common prediction c_i as its pseudo-label. The ranking score is not used as a loss weight.

B.2 Disagreement Scorers and Score Transformations

Task definitions. For a disagreement observation i, the two models propose different labels, so exactly one model can be correct or neither is correct. We define the trust target as

$$y_T(i) = \mathbb{I}\left[\hat{y}_i^G = y_i \ \vee \ \hat{y}_i^L = y_i\right]. \tag{35}$$

All labeled disagreement observations contribute to trust estimation. Preference supervision is restricted to observations with $y_T(i) = 1$:

$$y_P(i) = \mathbb{I}[\hat{y}_i^G = y_i], \qquad y_T(i) = 1. \tag{36}$$

Thus, the positive preference class denotes a correct graph prediction, and the negative class denotes a correct language prediction. Observations on which both models fail are excluded from preference supervision.

Selected configurations. Table 5 summarizes the selected trust and preference scorers. Each scorer uses its own feature configuration and fitted transformation. Source training uses the other four datasets; target training uses only the target dataset; pooled training combines source and target observations; transfer training performs source pretraining followed by target adaptation.

Classical classifiers. The Cora trust scorer is an RBF-kernel support vector classifier with $C_{\mathrm{SVM}} = 10$ and `gamma=scale`. Its input combines compact evidence with fixed graph and language projections through the `embed` mode. The Cora preference scorer is an L2-regularized logistic regression classifier with $C_{\mathrm{LR}} = 0.01$ using `context` features.

The Citeseer preference scorer is an L2-regularized logistic regression classifier with $C_{\mathrm{LR}} = 0.003$ using `embed` features.

Dataset	Task	Model	Features	Training	Dim.
Cora	Trust	RBF-SVM	`embed`	Pooled	47
	Preference	Logistic regression	`context`	Source	27
Citeseer	Trust	Neural network	Tabular + projected LLM	Transfer	86
	Preference	Logistic regression	`embed`	Source	47
PubMed	Trust	Extra Trees	`context`	Pooled	26
	Preference	Logistic regression	`compact`	Source	23
Arxiv	Trust	Logistic regression	`prob`	Target	21
	Preference	Joint neural network	`compact`	Transfer	23
Products	Trust	Logistic regression	`prob`	Source	41
	Preference	Joint neural network	`embed`	Transfer	47

Table 5: final disagreement scorers. Dimensions are measured after the selected feature transformation. For joint neural models, only the preference output is used in the final router, paired with the separately selected trust scorer.

The PubMed trust scorer is an Extra Trees classifier with 200 trees, maximum depth 8, minimum leaf size 12, and `max_features=1.0`. It uses `context` features. The PubMed preference scorer is an L2-regularized logistic regression classifier with $C_{\mathrm{LR}} = 0.01$ using `compact` features and the gold-fitted rank transformation.

The Arxiv trust scorer is an L2-regularized logistic regression classifier with $C_{\mathrm{LR}} = 0.01$ using probability features. The Products trust scorer is an L2-regularized logistic regression classifier with $C_{\mathrm{LR}} = 0.1$, also using probability features.

These classical configurations use random seed 42. Their fitted feature transformations follow Section A.5. Among these final disagreement scorers, only the PubMed preference classifier uses the rank transformation.

Joint neural training for Arxiv and Products. The selected Arxiv and Products preference scorers are obtained from networks trained jointly on trust and preference. For a transformed input $\boldsymbol{\xi}_i \in \mathbb{R}^d$, the network has the architecture

$$d \xrightarrow{\text{Linear+GELU+Dropout}_{0.15}} 32 \xrightarrow{\text{Linear+GELU}} 16 \xrightarrow{\text{Linear}} 2. \tag{37}$$

The two outputs are trust and preference logits. The input dimension is $d = 23$ for Arxiv and $d = 47$ for Products.

At each optimization step, separate batches are sampled for two tasks: $\mathcal{B}_T$ contains labeled disagreement observations, whereas $\mathcal{B}_P$ contains only observations with $y_T(i) = 1$. The objective is

$$\begin{aligned}\mathcal{L}_{\text{joint}} = &\frac{1}{|\mathcal{B}_T|} \sum_{i \in \mathcal{B}_T} \text{BCEWithLogits}\left(z_T(i), y_T(i)\right) \\ &+ \frac{1}{|\mathcal{B}_P|} \sum_{i \in \mathcal{B}_P} \text{BCEWithLogits}\left(z_P(i), y_P(i)\right).\end{aligned} \tag{38}$$

Each task uses a batch of 128 observations sampled with replacement according to its training weights. Both-wrong observations update the shared representation through the trust loss only.

Both configurations use 280 source-pretraining steps at learning rate 0.002, followed by 60 target-adaptation steps at learning rate 0.0005. Optimization uses AdamW, gradient-norm clipping at 5, and weight decay 0.01 for Arxiv and 0.05 for Products. Three networks are trained with seeds 42, 137, and 2026. Their preference logits are averaged to obtain the final raw preference score:

$$f_P(\mathbf{x}_i) = \frac{1}{3} \sum_{s \in \{42,137,2026\}} z_P^{(s)}(i). \tag{39}$$

The corresponding neural trust outputs are not used in the final router. Instead, the preference scores are paired with the independently selected logistic trust scorers in Table 5.

Training populations and dataset weights. Training observations are generated from the clean and augmented gold views. Branch membership and task targets are recomputed from each environment's candidate predictions. Within a dataset and task, eligible clean and augmented observations receive equal weight.

For source-only training, the four source datasets receive equal total mass. For pooled Cora and PubMed trust training, the target receives mass 0.25 and each source receives mass 0.1875. Target-only Arxiv trust training assigns all mass to Arxiv.

For transfer training, source and target observations are sampled in separate stages. Citeseer trust and Products joint training use equal source-dataset masses. Arxiv joint training uses source masses proportional to $1:1:2:2$ for Cora, Citeseer, PubMed, and Products, respectively. Its target dataset receives mass 0.25 in the combined statistics used for feature fitting and training-score normalization, with the remaining mass distributed across sources in those proportions. Products uses target mass 0.25 and source mass 0.1875 per dataset for these combined statistics. During target adaptation, sampling is restricted to the target dataset.

For single-task scorers, task eligibility is applied before fitting the input transformation. For jointly trained networks, the shared transformation is fitted using the training disagreement population, including both-wrong observations. Preference supervision nevertheless remains restricted to observations with a correct candidate. Node and view identifiers are not used as features.

Raw score conventions. Let $z_h(i) = f_h(\mathbf{x}_i)$ for $h \in \{T, P\}$. Logistic regression and SVM scorers return their decision functions, and neural scorers return logits. The PubMed trust tree ensemble returns transformed positive-class probabilities:

$$z_T(i) = \log \frac{\bar{\pi}_T(i)}{1 - \bar{\pi}_T(i)}, \qquad \bar{\pi}_T(i) = \min\big(1 - 10^{-6}, \max(10^{-6}, \pi_T(i))\big). \tag{40}$$

These raw scores are mapped to trust and preference estimates through

$$t_i = \sigma(a_T z_T(i) + b_T), \qquad r_i = \sigma(a_P z_P(i) + b_P). \tag{41}$$

The affine coefficients combine a normalization or calibration fitted on labeled observations with the validation-selected post-processing described below.

Gold out-of-fold calibration. The Citeseer and Products trust scorers use an affine calibration fitted on out-of-fold scores of the labeled target observations. Gold nodes are assigned to five folds, with all augmented views of the same original node kept in the same fold. For each held-out fold, the scorer is fitted using the remaining labeled target observations together with any source observations required by its training recipe. The held-out scores are then collected for calibration.

Let $\widehat{z}_j$ denote an out-of-fold score. The calibration parameters minimize

$$\sum_j \mathrm{BCEWithLogits}\left(a_h^{(0)} \widehat{z}_j + b_h^{(0)}, y_{j,h}\right) + \frac{\lambda}{2}\left[(a_h^{(0)} - 1)^2 + (b_h^{(0)})^2\right], \tag{42}$$

with $\lambda_{cal} = 2$ and $a_h^{(0)} \in [0.05, 20]$. If either binary class contains fewer than five calibration observations, the slope is fixed to one and only the bias is fitted.

The prototype features of Appendix A.4 are not recomputed within folds.

Training-score normalization. The remaining trust and preference scorers are normalized with their scores on the labeled training observations. Let $\omega_{j,h}$ denote normalized task-specific training weights, with $\sum_j \omega_{j,h} = 1$. Define

$$\begin{aligned} \mu_h &= \sum_j \omega_{j,h} z_h(j), \\ s_h &= \max\left(\sqrt{\sum_j \omega_{j,h}(z_h(j) - \mu_h)^2}, 0.1\right). \end{aligned} \tag{43}$$

Dataset	a_T	b_T	a_P	b_P
Cora	0.506661	1.364681	0.339507	-0.876366
Citeseer	0.125000	-0.999937	0.934017	-2.176022
PubMed	0.176727	0.627705	7.488961	-23.752220
Arxiv	1.160239	3.231565	0.261817	-1.088684
Products	0.240000	-0.123534	0.188425	-0.758166

Table 6: Affine coefficients of the disagreement score transformations (rounded to six decimal places).

The initial affine coefficients are

$$a_h^{(0)} = \frac{1}{s_h}, \qquad b_h^{(0)} = -\frac{\mu_h}{s_h}. \tag{44}$$

This operation standardizes the training-score scale; it does not itself fit a probability calibration model. For joint networks, trust and preference normalization use their respective eligible observations and weights.

Validation post-processing and final coefficients. Subsequent post-processing choices are selected on validation data. An additional scale γ_h and offset δ_h can be composed with the initial transformation:

$$a_h = \gamma_h a_h^{(0)}, \qquad b_h = \gamma_h b_h^{(0)} + \delta_h. \tag{45}$$

Table 6 reports the resulting coefficients, which include this post-processing and are applied directly to the raw scores.

B.3 Preference Routing and Selective Acceptance

Preference score transformation. The preference classifier returns a scalar score $f_P(\mathbf{x}_i)$. The final preference estimate is

$$r_i = \sigma(a_P f_P(\mathbf{x}_i) + b_P)\,. \tag{46}$$

The coefficients a_P, b_P combine the normalization fitted on labeled observations with the validation-selected post-processing. Their construction and dataset-specific values are given in Appendix B.2. The raw score can be a logistic decision function or an averaged neural logit, depending on the selected classifier. Applying the sigmoid gives a value in $(0, 1)$ but does not by itself establish probability calibration.

Routing boundary and tie convention. All final datasets use the preference boundary $\tau = 0.5$. The pseudo-label and disagreement selection score are computed as

$$(\widetilde{y}_i, s_D(i)) = \begin{cases} (\hat{y}_i^G, t_i r_i), & r_i \geq 0.5, \\ (\hat{y}_i^L, t_i(1 - r_i)), & r_i < 0.5. \end{cases} \tag{47}$$

For $t_i > 0$, this is equivalent to selecting the larger of $q_G(i)$ and $q_L(i)$. A preference tie at $r_i = 0.5$ is resolved in favor of the graph modality.

The neither-correct score $q_N(i) = 1 - t_i$ is not included in the modality-selection argmax and does not independently trigger rejection. Instead, low trust reduces the selected modality's score through multiplication by t_i. Rejection is determined by the subsequent acceptance budget.

Branch-specific acceptance budgets. For branch $B \in \{A, D\}$, the final budget is

$$k_B = \lceil \rho_B |\mathcal{U}_B| \rceil\,. \tag{48}$$

Nodes are ranked by descending branch score, with score ties resolved by node ID. An empty branch contributes no accepted nodes. The ratios selected for every dataset, label budget, and seed are listed in Table 7.

Table 7: Validation-selected acceptance ratios ρ_A/ρ_D (%) for every dataset, label budget k, and seed. Ratios are selected independently for each split from a finite grid, using validation labels only (Appendix B.3).

		Seed 42		Seed 43		Seed 44	
Dataset	k	ρ_A	ρ_D	ρ_A	ρ_D	ρ_A	ρ_D
Cora	3	25	15	40	12.5	75	17.5
	5	35	17.5	7.5	20	50	22.5
	10	50	15	50	22.5	40	22.5
Citeseer	3	35	17.5	25	17.5	10	17.5
	5	22.5	15	10	15	10	15
	10	22.5	22.5	35	20	10	25
PubMed	3	60	15	20	25	12.5	35
	5	50	20	25	20	25	17.5
	10	60	50	25	25	50	40
Arxiv	3	1	1.5	0.25	3.5	100	5
	5	0.25	0.25	0.5	0.75	0.5	0.5
	10	3.5	0.5	0.5	25	0.5	1
Products	3	40	7.5	100	30	100	10
	5	25	7.5	15	7.5	30	7.5
	10	20	15	22.5	12.5	17.5	7.5

Validation selection. Acceptance ratios are selected separately for each dataset and branch from a finite grid. For each ratio, the same ranking and rounding procedure is applied to the corresponding validation population, and must accept at least 20 validation observations. Among eligible policies, selection prioritizes accepted pseudo-label precision, followed by its Wilson lower confidence bound and then coverage. The selected ratio is subsequently applied to the inference population; the validation score cutoff itself is not reused as a fixed threshold. No test labels are used to select these acceptance ratios.

Pseudo-label output. For accepted agreement nodes, the output label is $\widetilde{y}_i = \hat{y}_i^G = \hat{y}_i^L$. For accepted disagreement nodes, it is the prediction selected by the preference boundary. The accepted node-label pairs are passed to the downstream learner. Routing scores are used only for selection, not as loss weights.

B.4 Multi-View Augmentation

To expose the router to diverse model behaviors under limited supervision, we construct semantic and structural perturbations of the few-shot support. Let $\mathcal{S}$ denote the support set and $\mathcal{A}$ the set of views comprising the clean input and its augmented views. For node i in environment e, the frozen models produce distributions $p_{i,e}^G$ and $p_{i,e}^L$ with predictions

$$\hat{y}_{i,e}^G = \arg\max_c p_{i,e}^G(c), \qquad \hat{y}_{i,e}^L = \arg\max_c p_{i,e}^L(c). \tag{49}$$

Each observation retains the original gold label y_i, while its evidence representation $\mathbf{x}_{i,e}$ is constructed from the environment-specific outputs and the corresponding context. The augmented support is

$$\mathcal{S}^{\text{aug}} = \{(\mathbf{x}_{i,e}, y_i) : (i,e) \in \mathcal{S} \times \mathcal{A}\}. \tag{50}$$

To make supervision reflect changes in model behavior, we recompute branch membership and routing targets for every observation. Let $a_{i,e} = \mathbb{I}[\hat{y}_{i,e}^G = \hat{y}_{i,e}^L]$ denote agreement. The task-specific targets are

$$\begin{aligned} y_A(i,e) &= \mathbb{I}[\hat{y}_{i,e}^G = y_i], && a_{i,e} = 1, \\ y_T(i,e) &= \mathbb{I}[\hat{y}_{i,e}^G = y_i \ \vee \ \hat{y}_{i,e}^L = y_i], && a_{i,e} = 0, \\ y_P(i,e) &= \mathbb{I}[\hat{y}_{i,e}^G = y_i], && a_{i,e} = 0,\ y_T(i,e) = 1. \end{aligned} \tag{51}$$

Different views of the same node can thus provide supervision for different routing tasks, allowing the router to learn how changes in semantic or structural evidence relate to correctness. Model parameters remain frozen throughout router training, and augmentation expands the observations of model behavior without increasing the number of independently labeled nodes.

Augmentation views. The augmentation pool comprises six views: clean input, 50% text truncation, mixed 30% text masking, 30% contiguous text-span deletion, 30% random edge deletion, and mixed 50% ego-edge masking. In mixed views, perturbations are applied only to the assigned nodes, while the remaining nodes retain their clean inputs.

Each observation (i, e) retains the original gold label y_i. Its agreement or disagreement membership and task-specific supervision are recomputed from the environment-specific candidate predictions. Text perturbations use shifted language representations together with clean graph representations; graph perturbations use shifted graph representations together with clean language representations. Both models remain frozen.

Task-specific training observations. To train the router on the augmented model behaviors, we construct an eligible observation set $\mathcal{S}_{d,h}$ for each dataset d and task $h \in \{A, T, P\}$. Each element (i, e) denotes node i observed in environment e. Agreement training includes observations on which the models agree; trust training includes all disagreement observations; preference training includes only disagreement observations with at least one correct candidate. The task target $y_h(i, e)$ is computed from the environment-specific predictions and the original gold label y_i. Augmentation therefore increases the number of observed model behaviors without increasing the number of independently labeled nodes.

Observation weights and supervised objectives. Let $\alpha_{d,h}$ denote the relative training mass assigned to dataset d for task h in the applicable training stage. Within each dataset and task, all eligible observations receive equal weight:

$$w_h(i, e) = \frac{\alpha_{d,h}}{|\mathcal{S}_{d,h}|}, \qquad (i, e) \in \mathcal{S}_{d,h}. \tag{52}$$

Thus, dataset mass controls the contribution of each dataset, while clean and augmented observations have equal weight within that dataset. Augmented views whose predictions are unchanged are kept with the same weight.

For scorers trained with binary cross-entropy, the weighted supervised component is

$$\mathcal{L}_h = \sum_{d:\,|\mathcal{S}_{d,h}|>0} \frac{\alpha_{d,h}}{|\mathcal{S}_{d,h}|} \sum_{(i,e)\in\mathcal{S}_{d,h}} \mathrm{BCEWithLogits}(f_h(\mathbf{x}_{i,e}), y_h(i, e)), \tag{53}$$

where $\mathbf{x}_{i,e}$ is the evidence representation for observation (i, e). This expression describes the supervised data-fitting term; regularization and optimization follow the selected classifier configuration. Tree ensembles and the RBF-SVM use their respective fitting objectives with the applicable observation weights. Fixed agreement rules require no classifier training.

Training scope and joint optimization. Depending on the selected configuration, training uses the target dataset, the other four source datasets, a weighted pool of source and target observations, or source pretraining followed by target adaptation. For staged training, the eligible datasets and their relative masses are defined separately for each stage.

Joint neural configurations use a shared trunk with trust and preference outputs. Separate task-specific samplers implement the corresponding observation weights, and training minimizes

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_T + \mathcal{L}_P. \tag{54}$$

Both-wrong observations contribute through the trust loss only. In the final Arxiv and Products configurations, the jointly trained preference output is paired with a separately selected trust scorer. Dataset-specific model architectures, training scopes, and optimization settings are detailed in Appendices B.1 and B.2.

Data used for fitting and selection. Router parameters and fitted input transformations use gold training observations only. Score transformations use gold-derived calibration or training-score normalization, followed by validation-based selection of the applicable post-processing and routing choices. Where gold out-of-fold calibration is used, all views of the same original node are assigned to the same fold. Node and view identifiers are used only to group folds and are never used as features. Score-transformation procedures and acceptance-policy selection are detailed in Appendices B.2 and B.3.

C Implementation Details for Residual Classifier

C.1 Multi-modal Residual Classifier

Architecture. For node i, we independently transform the frozen representations h_i^G and h_i^L into a common dimensionality:

$$\begin{aligned} u_i^G &= \mathrm{GELU}\big(W_G\,\mathrm{LN}_G(h_i^G) + b_G\big)\,, \\ u_i^L &= \mathrm{GELU}\big(W_L\,\mathrm{LN}_L(h_i^L) + b_L\big)\,, \end{aligned} \tag{55}$$

where LN_G and LN_L are separate layer-normalization modules and $u_i^G, u_i^L \in \mathbb{R}^{d_R}$. These trainable projections belong to the residual module and are optimized through its classification objective. The concatenated projections pass through a shared multilayer perceptron to give the residual distribution

$$p_i^R = \mathrm{softmax}\big(W_2\,\mathrm{GELU}\big(W_1[u_i^G; u_i^L] + b_1\big) + b_2\big)\,. \tag{56}$$

The hidden layer has width 64, and d_R is selected on validation data. The residual classifier predicts over all C classes.

Fusion. The scalars α and β are shared across nodes and classes within a model and are initialized to one, giving equal initial weight to all three distributions. Their normalization also determines the residual contribution, whose unnormalized coefficient is fixed to one. The mixture is evaluated in log space for numerical stability.

Supervision pools. Let $\mathcal{P}$ be the accepted set produced by the router. We distinguish accepted agreement nodes from accepted disagreement nodes according to the selected modality:

$$\begin{aligned} &\mathcal{P}_A = \mathcal{P} \cap \mathcal{U}_A, \\ &\mathcal{P}_D^E = \big\{ i \in \mathcal{P} \cap \mathcal{U}_D : \tilde{y}_i = \hat{y}_i^E \big\}\,, \qquad E \in \{G, L\}. \end{aligned} \tag{57}$$

Since the models disagree on these nodes, the two source pools are disjoint.

Losses. For a collection of predictive distributions $q = \{q_i\}$, the support and disagreement losses are

$$\begin{aligned} \mathcal{L}_{\mathrm{sup}}(q) &= -\frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} \log q_i(y_i), \\ \mathcal{L}_{\mathrm{dis}}(q) &= -\frac{1}{2} \sum_{\substack{E \in \{G,L\} \\ |\mathcal{P}_D^E| > 0}} \frac{1}{|\mathcal{P}_D^E|} \sum_{i \in \mathcal{P}_D^E} \log q_i(\tilde{y}_i). \end{aligned} \tag{58}$$

Each source is averaged separately and receives a coefficient of $1/2$, preventing a larger source pool from dominating solely through its size. The primary objective applies these losses to the fused distribution p^J, while auxiliary terms directly supervise the residual distribution p^R.

On accepted agreement nodes, we use the mean distribution $m_i = \frac{1}{2}(p_i^G + p_i^L)$ as a fixed soft target:

$$\mathcal{L}_{\mathrm{keep}} = \frac{1}{|\mathcal{P}_A|} \sum_{i \in \mathcal{P}_A} D_{\mathrm{KL}}\big(m_i \,\|\, p_i^J\big)\,, \tag{59}$$

with $\mathcal{L}_{\mathrm{keep}} = 0$ when $\mathcal{P}_A$ is empty. This optional term discourages unnecessary changes to consensus predictions while retaining their uncertainty over alternative classes. The full objective is

$$\begin{aligned} \mathcal{L}_{\mathrm{residual}} = \mathcal{L}_{\mathrm{sup}}(p^J) &+ \lambda_D \mathcal{L}_{\mathrm{dis}}(p^J) \\ &+ \lambda_{R,S} \mathcal{L}_{\mathrm{sup}}(p^R) + \lambda_D \lambda_{R,D} \mathcal{L}_{\mathrm{dis}}(p^R) \\ &+ \lambda_A \mathcal{L}_{\mathrm{keep}}, \end{aligned} \tag{60}$$

where all coefficients are nonnegative. The two auxiliary coefficients $\lambda_{R,S}$ and $\lambda_{R,D}$ independently control direct residual supervision from gold labels and pseudo-labels, and setting a coefficient to zero disables the corresponding optional term.

Prediction. The selected model applies the fusion equation to every query node, including those excluded from pseudo-label supervision:

$$\hat{y}_i^J = \arg \max_{c \in \{1,\dots,C\}} p_i^J(c). \tag{61}$$

C.2 Configurations and Details

Trainable parameters and sampling. We optimize the projection modules, residual classifier, and trainable fusion coefficients using AdamW. The representations and predictive distributions are treated as fixed inputs, and gradients do not propagate into either modality. Router decisions and accepted pseudo-labels also remain fixed throughout residual training.

At each optimization step, independent minibatches are sampled uniformly with replacement from the active supervision pools: the labeled support $\mathcal{S}$, the accepted disagreement pools $\mathcal{P}_D^G$ and $\mathcal{P}_D^L$, and, when agreement preservation is enabled, the accepted agreement pool $\mathcal{P}_A$. Each pool contributes through its corresponding loss term in Equation 60.

Fusion optimization variants. Validation-selected configurations can assign the fusion coefficients a learning rate separate from that of the residual network. Other options include an initial period during which the fusion coefficients remain fixed and restricting their updates to gradients from gold supervision. These restrictions apply to fusion-coefficient optimization; the residual network continues to use its configured supervision terms. A fixed-fusion variant keeps $\alpha = \beta = 1$ throughout training, yielding

$$p_i^J = \frac{p_i^G + p_i^L + p_i^R}{3}. \tag{62}$$

Optional class weighting. For imbalanced training pools, an optional configuration replaces the mean in each classification loss with a normalized class-weighted mean. For a supervision pool $\mathcal{B}$ with hard targets t_i, this takes the form

$$\mathcal{L}_{\mathrm{cw}}(q; \mathcal{B}) = -\frac{\sum_{i \in \mathcal{B}} \eta_{t_i} \log q_i(t_i)}{\sum_{i \in \mathcal{B}} \eta_{t_i}}, \tag{63}$$

where η_c is the weight assigned to class c. Class weights are derived from smoothed inverse frequencies computed only from support labels and accepted disagreement pseudo-labels. For disagreement supervision, weighting is applied within each modality-selected pool, retaining the source coefficients in Equation 58. The agreement-preservation KL loss is unchanged.

Configuration and checkpoint selection. Configurations and complete model checkpoints are selected independently within each data split. Selection prioritizes validation accuracy, with validation negative log-likelihood breaking ties. The selected checkpoint includes the projection modules, residual classifier, and fusion parameters, preserving their jointly selected state.

Transductive prediction and data boundaries. The selected model evaluates the fused distribution in Equation 5 for every query node, including nodes that were not accepted for pseudo-label supervision. Query representations and accepted pseudo-labels are used transductively. Query ground-truth labels are excluded from optimization, class-weight estimation, configuration selection, and checkpoint selection.

D Experimental Details

D.1 Datasets and Splits

Table 8 summarizes the datasets. Cora, Citeseer, and PubMed are citation networks in which nodes are papers described by their titles and abstracts, and edges are citations. Arxiv is the OGB citation

graph of computer science papers, and Products is the commonly used subset of the OGB Amazon co-purchasing graph, whose nodes are products described by their titles and descriptions. Cornell and Wisconsin are webpage graphs from the WebKB collection with low homophily ratios (0.124 and 0.181, respectively), and we use the text-attributed versions released by Wang et al. (2025).

We follow the split protocol of GNN-as-Judge (Xu & Ding, 2026) and APPNP (Klicpera et al., 2019). Under the k-shot setting with $k \in \{3, 5, 10\}$, the support set $\mathcal{S}$ contains $\min(|\mathcal{V}_c|, k)$ labeled nodes per class. For Cora, Citeseer, and PubMed, we use 500 validation nodes and randomly select 1,000 test nodes from the remaining nodes. For Arxiv and Products, the support set is sampled from the official training split, and the official validation and test splits are used. Each experiment is repeated with three random seeds, which change the support set and the model initialization. All nodes outside $\mathcal{S}$ constitute the unlabeled set $\mathcal{U}$, on which the router operates.

Table 8: Statistics of the evaluated datasets. h denotes the edge homophily ratio of the heterophilous graphs.

Dataset	Domain	#Nodes	#Edges	#Classes	Val / Test
Cora	Citation	2,708	5,429	7	500 / 1,000
Citeseer	Citation	3,327	4,732	6	500 / 1,000
PubMed	Citation	19,717	44,338	3	500 / 1,000
ogbn-arxiv	Citation	169,343	1,166,243	40	OGB official
ogbn-products (subset)	E-commerce	54,025	74,420	47	OGB official
Cornell (h=0.124)	Webpage	191	274	5	36 / 105
Wisconsin (h=0.181)	Webpage	265	459	5	52 / 167

D.2 Baselines

GNNs. *GCN* (Kipf & Welling, 2017) aggregates normalized neighborhood features layer by layer, and *SGC* (Wu et al., 2019) removes the nonlinearities of GCN and precomputes the propagated features.

Graph–LLM models. *GLEM* (Zhao et al., 2023) alternately trains a language model and a GNN through a variational EM framework. *LLaGA* (Chen et al., 2024a) encodes node neighborhoods into structure-aware token sequences and projects them into the LLM's input space. *GraphGPT* (Tang et al., 2024) aligns graph structure with the LLM through graph instruction tuning.

Pseudo-labeling methods. *MoDis* (Pei et al., 2024) and *TA-GST* (Liu et al., 2026a) are GNN self-training methods that select GNN pseudo-labels according to training dynamics and topological consistency, respectively. *LLM-GNN* (Chen et al., 2024b) selects representative and diverse nodes for LLM annotation and trains a GNN on the annotated nodes, and *DMA* (Sheng et al., 2025) incorporates the estimated class-wise LLM noise distribution into active node selection. *Locle* (Zhang et al., 2025) iteratively selects critical nodes with a GNN and refines their pseudo-labels with LLM predictions over a rewired topology. *GNN-as-Judge* (Xu & Ding, 2026) selects influential unlabeled nodes, uses the GNN as the judge on disagreement nodes, and fine-tunes the LLM with instruction tuning on agreement nodes and preference tuning on disagreement nodes.

Reproduction of GNN-as-Judge. We rerun GNN-as-Judge with its official implementation under our splits and seeds. The released code contains an additional GNN retraining step (function `retrain_gnn`) that is not described in the original paper and that uses the ground-truth labels of unlabeled nodes. Since these labels are unavailable under the few-shot protocol and include those of the evaluation nodes, we disable this step and keep all other components and steps of the official implementation unchanged. All GNN-as-Judge results in this paper are obtained with this reproduction and may therefore differ from those reported by Xu & Ding (2026).

The results of GCN, SGC, GLEM, LLaGA, and GraphGPT are taken from GNN-as-Judge, which uses the same splits and evaluation protocol. The pseudo-labeling baselines are rerun under our splits and seeds with their official implementations and recommended hyperparameters, using LLaMA3-8B wherever an LLM is required.

D.3 Implementation Details

Frozen models. The GNN expert produces class distributions p^G and 128-dimensional representations h^G; we use a two-layer GCN with 128-dimensional hidden and output representations, LayerNorm, ReLU, and dropout of 0.3 between the layers. The encoder is pretrained without labels for 200 epochs using AdamW with a learning rate of 10^{-3} and weight decay of 10^{-4}. Pretraining combines masked-feature reconstruction and inner-product-based edge prediction, with additional cross-view cosine alignment and variance regularization on all datasets except ogbn-arxiv. The corresponding loss weights are $(1, 0.2, 0.5, 0.05)$, while ogbn-arxiv uses $(1, 0.5, 0, 0)$. After pretraining, the encoder is frozen, and a two-layer MLP classification head is trained on the few-shot labeled nodes using cross-entropy. Head training uses AdamW with weight decay of 10^{-4}, a learning rate of 10^{-3} (5×10^{-4} for PubMed), and at most 800 epochs for Citeseer and ogbn-arxiv or 1,000 epochs for the other datasets, with early-stopping patience of 150 and 200 epochs, respectively. Checkpoints are selected by validation accuracy, with validation loss breaking ties. The encoder outputs h^G, and softmax-normalized MLP logits yield p^G. The LLM expert is LLaMA3-8B by default, with Qwen3-8B and Mistral-7B used in Appendix I; its class distribution p^L is obtained by prompting the frozen LLM with the node text and a dataset-specific list of numbered categories, instructing it to return only the category number, and applying a softmax over the next-token logits of the corresponding single-token category numbers. Specifically, $p^L_c = \exp(z_{\tau(c)}) / \sum_{j=1}^{C} \exp(z_{\tau(j)})$, where $z_{\tau(c)}$ is the vocabulary logit of the number token assigned to class c. The prompt is formatted using the model's chat template, with a maximum input length of 2,048 tokens, and $h^L \in \mathbb{R}^{4096}$ is its hidden representation of the node text. We use prompt:

```
{node_text}
{dataset_specific_instruction}

Category number options:
1. {category_1}
2. {category_2}
...
{C}. {category_C}

Reply only one category number from 1 to {C}.

Do not output the category name or any explanation words.

Answer:
```

For the heterophilous graphs, H2GCN (Zhu et al., 2020) serves as the GNN expert. Neither expert is updated by FairRouter.

Router. The router uses the evidence of Appendix A and the dataset-specific scorers of Appendix B, trained on the six augmentation views of Appendix B.4. Acceptance ratios are selected on the validation set by prioritizing pseudo-label precision, then its Wilson lower bound, then coverage.

Residual classifier. The residual MLP has a hidden width of 64 and is optimized with AdamW (Loshchilov & Hutter, 2019). The learning rate is searched over $\{10^{-4}, 3\times10^{-4}, 10^{-3}, 3\times10^{-3}, 10^{-2}\}$, and the loss coefficients and projection dimension d_R are searched per split. Configurations and checkpoints are selected by validation accuracy, with validation negative log-likelihood as the tie-breaker (Appendix C.2).

Environment. All experiments are conducted on RTX4090.

D.4 Pseudo-Label Evaluation Protocol

For a set of accepted pseudo-labeled nodes $\mathcal{P}$ drawn from a candidate population $\mathcal{U}'$, we define

$$\text{Precision} = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} \mathbb{I}[\tilde{y}_i = y_i], \qquad \text{Coverage} = \frac{|\mathcal{P}|}{|\mathcal{U}'|}. \tag{64}$$

Ground-truth labels of unlabeled nodes are used only to compute these metrics.

Overall precision (Figures 3 and 8). For each method, $\mathcal{P}$ is the pseudo-label set that the method actually uses to train its final model, so the metric reflects the supervision each downstream learner receives. For FairRouter, $\mathcal{P}$ is the union of the accepted agreement and disagreement nodes.

Precision–coverage on disagreement nodes (Figure 10). Here $\mathcal{U}' = \mathcal{U}_D$, and each selector ranks disagreement nodes and accepts the top fraction at a requested coverage. Let $c_G = \max_c p_i^G(c)$ and $c_L = \max_c p_i^L(c)$.

- **FairRouter** assigns $\tilde{y}_i = \hat{y}_i^{e_i^\star}$ and ranks nodes by $s_D(i) = q_{e_i^\star}(i)$ (Section 3).
- **Confidence** assigns the label of the more confident expert and ranks nodes by $\max(c_G, c_L)$; this is the disagreement branch of the *Confidence router* ablation.
- **Confidence-based preference** keeps the learned trust score t_i but replaces r_i with $\mathbb{I}[c_G \geq c_L]$, i.e., it follows the more confident expert and ranks nodes by t_i.

Markers indicate the deployed operating points, where all selectors accept the same number of nodes as FairRouter under the validation-selected ratio ρ_D.

E Pseudo-Label Quality Analysis

E.1 Pseudo-Label Precision under Other Label Budgets

Figure 8 repeats the overall precision analysis of Section 4.3 under the 3-shot and 10-shot settings, following the protocol of Appendix D.4: for each method, precision is computed on the pseudo-labels that it actually uses to train its final model.

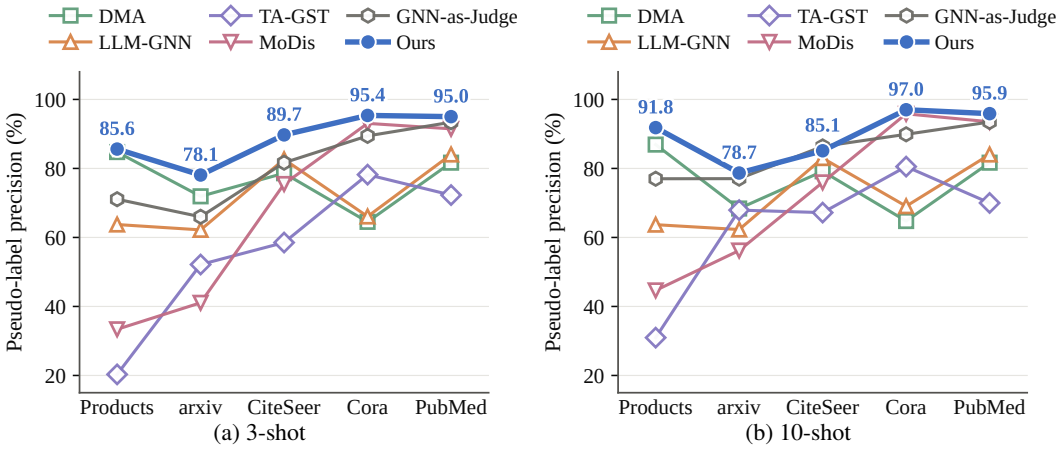


Figure 8: Precision of the pseudo-labels used by each method under the (a) 3-shot and (b) 10-shot settings. The 5-shot results are shown in Figure 3.

E.2 Branch-Level Pseudo-Label Statistics

Figure 9 and Table 9 decompose the pseudo-labels of FairRouter by routing branch under every label budget. Statistics are computed on the query population, i.e., all nodes outside the support and validation sets, following the protocol of Appendix D.4. Coverage is measured within each branch and equals, up to rounding, the acceptance ratio selected on the validation set for that split (Table 7).

Routed pseudo-labels multiply the available supervision. At 3 shots, where the gains of FairRouter in Table 1 are largest, the accepted set contains 29–577 times as many correct pseudo-labels as labeled nodes, at an overall precision of 78.1–95.4%. Across all budgets, the overall precision exceeds 88% in 11 of the 15 settings, and correct pseudo-labels outnumber wrong ones on every dataset, budget, and seed.

Agreement pseudo-labels are nearly clean; disagreement is where routing matters. Accepted agreement nodes reach 87.9–98.6% precision and are more precise than accepted disagreement

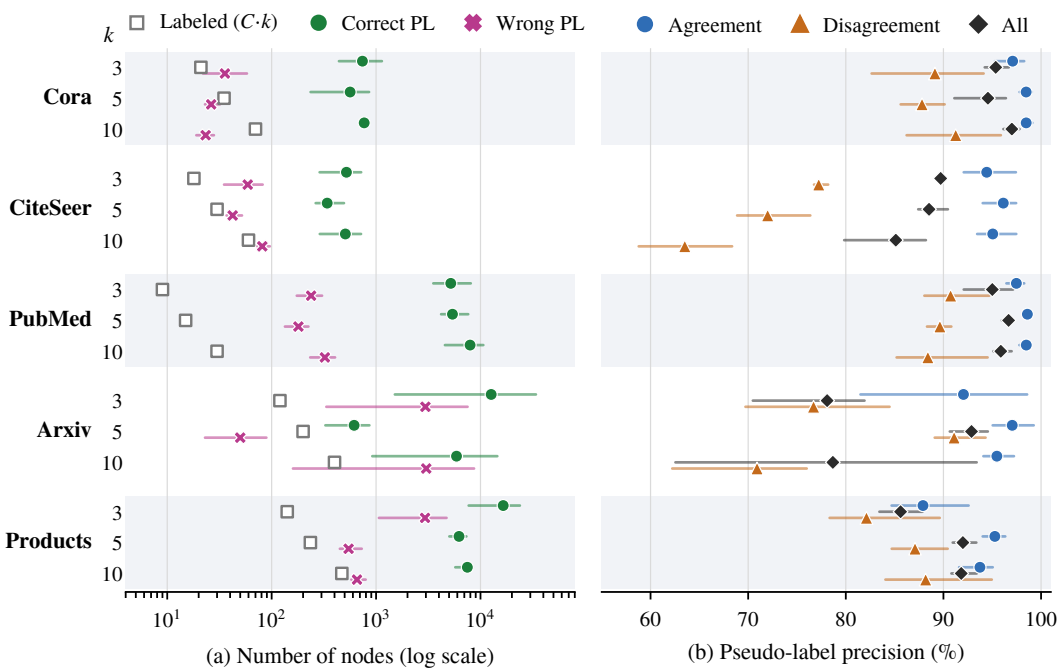


Figure 9: Pseudo-labels of FairRouter by routing branch under the 3-, 5-, and 10-shot settings. (a) Number of labeled nodes ($C \cdot k$), correct pseudo-labels, and wrong pseudo-labels (log scale). (b) Precision of the accepted agreement, disagreement, and all pseudo-labels. Markers show the mean over three seeds; horizontal lines show the range over seeds.

nodes in all 15 settings. The disagreement branch is harder by construction, since at most one of the two candidates can be correct; nevertheless, its accepted nodes reach 82.1–91.3% precision on Cora, PubMed, and Products. These are exactly the nodes on which a fixed-teacher method must side with one model, and they account for 15–91% of the accepted set. Removing them is the most harmful ablation under scarce labels (-2.94 pp at 3 shots; Section 4.4). Citeseer is the exception: its disagreement precision is the lowest among the five datasets (63.5–77.2%) and decreases as the budget grows, which makes this branch the main target for improving the router.

Downstream accuracy is robust to variation in pseudo-supervision. Because acceptance ratios are selected per split, the amount of pseudo-supervision varies across seeds, most strongly on Arxiv. At 3 shots, the three seeds accept 1,847, 3,560, and 41,483 pseudo-labels with overall precision between 70.5% and 81.9%; at 10 shots, seed 43 accepts 23,175 pseudo-labels at 62.6% precision. Nevertheless, the standard deviation of Arxiv accuracy in Table 1 remains 0.12 and 0.86 pp at 3 and 10 shots. This is consistent with the design of the residual classifier: the disagreement loss averages each modality-selected pool separately (Eq. 58), and the fused prediction stays anchored on the frozen expert distributions (Eq. 5), so noisy pseudo-labels can adjust but not override the experts.

Table 9: Branch-level statistics of the pseudo-labels of FairRouter (mean over three seeds; precision in %, ± population std). Cov.: accepted fraction of the branch population (%). #PL and #Correct: accepted and correct pseudo-labels. Ratio: #Correct divided by $C \cdot k$; since $C \cdot k \geq |\mathcal{S}|$, it is a lower bound on the gain in supervision.

		Agreement		**Disagreement**		**All**			
Dataset	k	Cov.	Prec.	Cov.	Prec.	#PL	#Correct	Ratio	Prec.
Cora	3	46.7	97.09±1.30	15.1	89.13±4.80	776	740	35×	95.36±1.02
	5	30.9	98.47±0.52	20.0	87.81±1.84	591	564	16×	94.56±2.43
	10	46.7	98.48±0.45	20.0	91.26±3.95	793	769	11×	97.01±0.68
Citeseer	3	23.4	94.43±2.23	17.6	77.23±0.68	579	520	29×	89.72±0.43
	5	14.2	96.13±1.51	15.1	72.00±3.18	383	340	11×	88.53±1.37
	10	22.5	95.05±1.73	22.6	63.51±3.91	589	508	8×	85.12±3.76
PubMed	3	30.8	97.47±0.76	25.0	90.73±2.86	5,432	5,193	577×	95.01±2.14
	5	33.3	98.60±0.27	19.2	89.64±1.02	5,571	5,391	359×	96.67±0.50
	10	45.0	98.49±0.50	38.3	88.41±4.33	8,274	7,951	265×	95.88±0.80
Arxiv	3	33.8	92.05±7.53	3.3	76.70±6.05	15,630	12,662	106×	78.09±5.38
	5	0.4	97.05±1.72	0.5	91.10±2.30	666	616	3×	92.88±1.64
	10	1.5	95.47±1.31	8.8	70.91±6.17	8,917	5,894	15×	78.67±12.63
Products	3	80.0	87.91±3.38	15.8	82.12±5.30	19,448	16,503	117×	85.61±1.80
	5	23.3	95.27±0.95	7.5	87.09±2.44	6,777	6,232	27×	91.99±1.02
	10	20.0	93.74±1.52	11.7	88.18±4.83	8,143	7,487	16×	91.83±1.13

F Precision–Coverage Analysis on Disagreement Nodes

Methods in Figure 3 accept different numbers of pseudo-labels, so their precision is not measured at equal coverage. To isolate the routing decision, Figure 10 compares selectors at *matched* coverage on disagreement nodes, where the two modalities conflict and the choice of modality matters most. The selectors are defined in Appendix D.4. Two reference points help to read the curves. At full coverage every disagreement node is accepted, so precision depends only on which modality is followed, i.e., on the preference decision. At low coverage, precision depends on how well the ranking concentrates correct candidates at the top, i.e., on the trust decision.

Raw confidence can be misleading. On Cora, both confidence-based selectors reach only about 45–52% precision at the deployed coverage, whereas FairRouter reaches about 88%. Even at full coverage, FairRouter is about 20 pp more precise (62% versus 41%), which isolates the contribution of the learned preference. The most confident expert is therefore frequently wrong on Cora's disagreement nodes. This supports Challenge 1: the confidences of heterogeneous models are not directly comparable, and they must be assessed jointly in a shared evidence space.

Trust determines where reliable supervision exists. On Arxiv, all selectors converge to about 42% precision at full coverage, meaning that many disagreement nodes have no correct candidate at all. Reliable pseudo-labels therefore exist only at the head of the ranking, where FairRouter is the most precise (about 91% versus 85% and 75% at the deployed point). This explains why validation selects disagreement acceptance ratios of only 0.25–0.75% on Arxiv at 5 shots (Table 7).

Confidence is competitive when one modality dominates. On PubMed and Products, plain confidence is about 4–6 pp more precise than FairRouter at the operating point. On these graphs, the LLM is the dominant expert (e.g., 84.47% by direct inference on PubMed), so its confidence is already informative, and a router fitted on five labels per class does not fully recover this signal. Two observations nevertheless favor the learned router. First, pairing the learned trust score with a confidence-based preference is never clearly better than FairRouter and is markedly worse on Cora, Arxiv, and Products; the learned trust should be paired with the preference head trained alongside it rather than with a heuristic. Second, the full model is more accurate than the confidence router on both datasets under every budget (Tables 10–12). Combining confidence priors with learned routing is a promising extension (Appendix K).

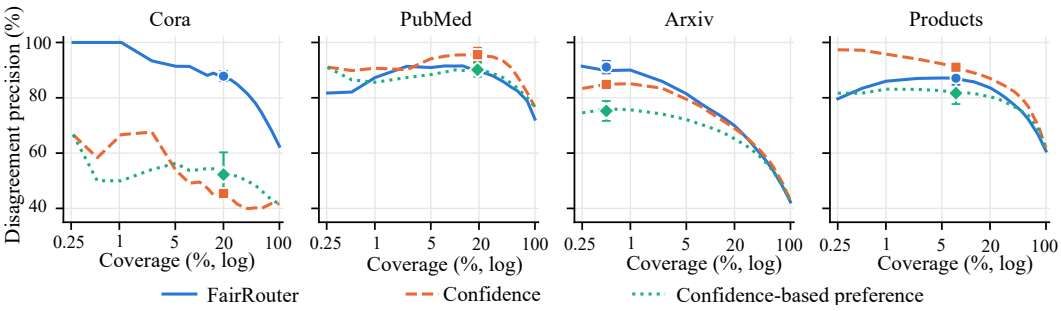


Figure 10: Precision–coverage on the disagreement branch (5-shot). Precision is the fraction of accepted nodes whose pseudo-label is correct; coverage (log scale) is relative to all disagreement nodes. *Confidence* follows and ranks by the more confident expert; *Confidence-based preference* keeps FairRouter's trust score but chooses the modality by confidence. Curves average three seeds at each requested coverage; markers show the deployed operating points (mean ± population std over seeds), at which all selectors accept the same number of nodes.

G Ablation Study Details

G.1 Ablation Settings

Let p^G and p^L denote the class probabilities of the frozen GNN and LLM models, and let p^R denote the prediction of the residual branch (Repair) computed from both models' representations. The full model outputs $p^J = (\alpha p^G + \beta p^L + p^R)/(\alpha + \beta + 1)$ with learnable positive scalars α, β, and is trained with gold labels on the support set, routed pseudo-labels on disagreement nodes, and a consistency constraint on agreement nodes, together with auxiliary losses on p^R where enabled. Retrained variants modify one component and retrain only the downstream modules.

Confidence router replaces the learned router with each expert's maximum class probability c_G, c_L: disagreement nodes take the label of the more confident expert and are ranked by $\max(c_G, c_L)$, agreement nodes are ranked by $\sqrt{c_G c_L}$, and the same number of nodes as in the full model is accepted in each group. **w/o disagreement pseudo-labels** removes the pseudo-label supervision on both p^J and p^R, while **Gold supervision only** additionally removes the consistency constraint, so that downstream training uses only the labeled support nodes. **G / L representation only** zeroes out the other expert's representation in the input of Residual; the corresponding expert probabilities are still fused, so these variants measure the contribution of representations rather than removing an expert. **Repair-only training** uses p^R as the output for both training and inference. **w/o repair auxiliaries** removes the losses applied directly to p^R; Repair Residual is still trained through p^J. Since these losses are enabled only in a subset of configurations, the variant coincides with the full model elsewhere.

The reference settings involve no retraining. **Equal expert fusion** predicts with $\frac{1}{2}p^G + \frac{1}{2}p^L$. **Validation-tuned expert fusion** predicts with $wp^G + (1-w)p^L$, where $w \in \{0, 0.05, \ldots, 1\}$ is selected on the validation set for each dataset, shot, and seed. **Residual readout** takes p^R from the trained full model. It differs from Residual-only training in that Residual was optimized as part of the fused output rather than as a standalone classifier.

G.2 Full Ablation Results

Tables 10–12 report per-dataset results under each label budget. Beyond the averages discussed in Section 4.4, the per-dataset results reveal three patterns.

The routing effect is dataset-dependent and follows the precision gaps. Replacing the learned router costs 6.70 pp on Cora but only 0.27 pp on PubMed at 3 shots. This matches Figure 10, where the learned router's precision advantage is largest on Cora and confidence is competitive on PubMed. Even on PubMed and Products, where confidence is more precise on the disagreement branch, the full model remains more accurate at every budget.

Structural and semantic representations are needed on different graphs. Removing the LLM representation hurts Arxiv (up to -4.10 pp) and PubMed (up to -2.97 pp) the most, whereas removing the GNN representation hurts Cora the most (-3.37 pp at 3 shots). The residual branch thus draws on whichever modality carries the discriminative signal on each graph.

Auxiliary residual losses matter on the large graphs. Removing the auxiliary losses on p^R costs 1.16/1.37/0.83 pp on average, and the entire drop comes from Arxiv (up to -4.57 pp) and Products (up to -2.37 pp), the only datasets on which these losses are enabled; on Cora and PubMed the variant coincides with the full model. With 40–47 classes, the residual branch needs direct supervision to learn a fine-grained decision boundary, which agrees with the sensitivity analysis in Figure 7(b).

The consistency constraint has a small, budget-dependent effect. The difference between *w/o disagreement pseudo-labels* and *Gold supervision only* is -0.24, $+0.30$, and 0.00 pp at 3, 5, and 10 shots. In the absence of disagreement supervision, therefore, the agreement consistency term alone neither helps nor hurts substantially.

Table 10: Core component ablations under the 3-shot setting. Columns 2–6 report 3-shot accuracy (%, mean ± population std over three seeds); the last three columns report the change in the equal-weight four-dataset average relative to Full (pp) at each label budget, with the current budget highlighted. All retrained variants reuse the per-split hyperparameters of Full.

	3-shot accuracy					Δ Avg. vs. Full		
Variant	Cora	PubMed	Arxiv	Products	Avg.	**3-shot**	5-shot	10-shot
Full (ours)	**79.37**±0.17	**90.80**±1.04	**58.13**±0.12	**72.30**±0.85	**75.15**	–	–	–
Routing								
Confidence router	72.67±0.76	90.53±0.25	55.60±0.64	70.90±1.63	72.42	−2.73	−2.43	−1.49
Supervision								
w/o disagreement pseudo-labels	73.93±0.95	89.07±1.60	56.90±0.71	68.93±0.53	72.21	−2.94	−2.61	−1.69
Gold supervision only	73.47±0.48	88.93±1.47	56.90±0.71	68.57±0.25	71.97	−3.18	−2.31	−1.69
Representation								
G representation only	78.10±1.20	87.83±2.00	54.03±1.68	69.50±2.05	72.37	−2.78	−2.81	−1.69
L representation only	76.00±1.53	90.27±1.11	57.37±1.73	71.97±0.76	73.90	−1.25	−0.91	−0.69
Output								
Repair-only training	77.53±0.80	89.63±1.76	55.80±2.00	69.00±0.85	72.99	−2.16	−3.23	−1.79
w/o repair auxiliaries	**79.37**±0.17	90.77±1.00	55.90±0.70	69.93±1.65	73.99	−1.16	−1.37	−0.83
Reference (no retraining of Full)								
Equal expert fusion	71.37±0.90	87.40±2.29	53.43±1.26	69.57±0.57	70.44	−4.71	−4.32	−4.62
Validation-tuned expert fusion	78.67±0.65	89.20±2.16	54.73±0.34	69.03±0.21	72.91	−2.24	−2.33	−2.43
Residual readout	76.47±3.37	90.37±1.16	52.47±1.89	65.37±4.23	71.17	−3.98	−5.72	−3.76

Table 11: Core component ablations under the 5-shot setting. Columns 2–6 report 5-shot accuracy (%, mean ± population std over three seeds); the last three columns report the change in the equal-weight four-dataset average relative to Full (pp) at each label budget, with the current budget highlighted. All retrained variants reuse the per-split hyperparameters of Full.

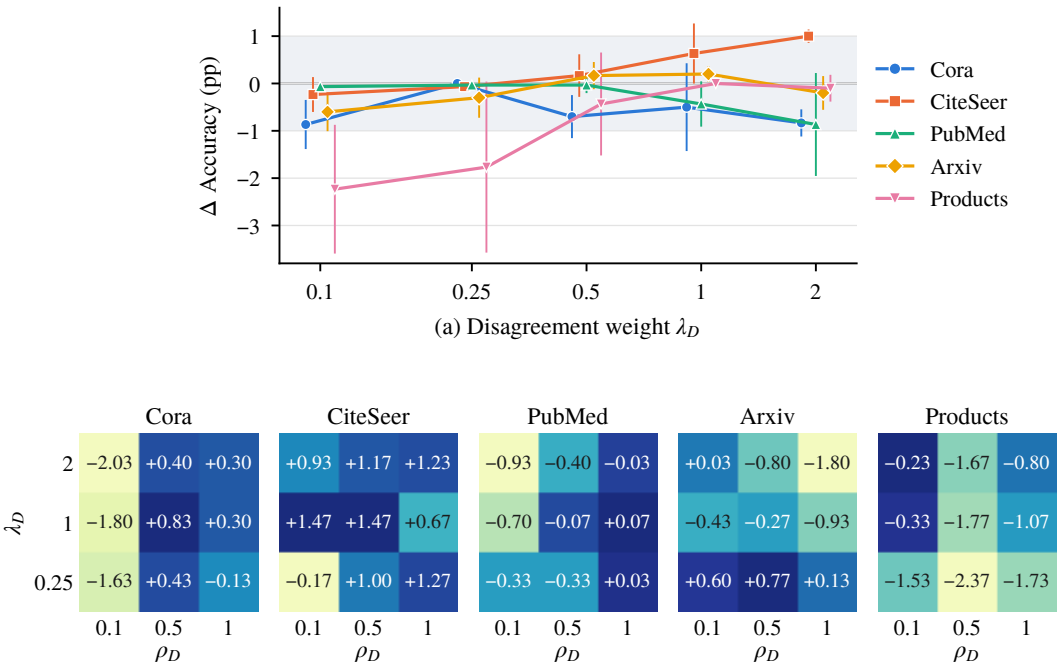
	5-shot accuracy					Δ Avg. vs. Full		
Variant	Cora	PubMed	Arxiv	Products	Avg.	3-shot	**5-shot**	10-shot
Full (ours)	**80.70**±1.91	**91.47**±1.30	**60.40**±1.56	**72.43**±0.41	**76.25**	–	–	–
Routing								
Confidence router	74.83±2.23	90.47±1.10	59.20±1.24	70.77±1.48	73.82	−2.73	−2.43	−1.49
Supervision								
w/o disagreement pseudo-labels	75.47±2.52	90.10±1.35	59.73±1.18	69.27±1.27	73.64	−2.94	−2.61	−1.69
Gold supervision only	76.67±0.83	90.23±1.25	59.60±1.07	69.27±1.27	73.94	−3.18	−2.31	−1.69
Representation								
G representation only	79.40±1.79	88.57±1.83	56.43±1.99	69.37±1.18	73.44	−2.78	−2.81	−1.69
L representation only	79.00±2.27	90.93±1.41	59.83±1.43	71.60±0.79	75.34	−1.25	−0.91	−0.69
Output								
Repair-only training	78.20±1.85	90.63±1.92	56.10±1.66	67.17±2.57	73.02	−2.16	−3.23	−1.79
w/o repair auxiliaries	**80.70**±1.91	**91.47**±1.30	55.83±2.43	71.53±0.29	74.88	−1.16	−1.37	−0.83
Reference (no retraining of Full)								
Equal expert fusion	72.77±1.53	89.67±2.59	55.60±2.20	69.70±1.28	71.93	−4.71	−4.32	−4.62
Validation-tuned expert fusion	79.90±1.94	90.07±2.08	56.13±1.96	69.60±0.29	73.92	−2.24	−2.33	−2.43
Residual readout	76.57±1.84	90.93±1.60	53.57±2.02	61.03±2.09	70.53	−3.98	−5.72	−3.76

Table 12: Core component ablations under the 10-shot setting. Columns 2–6 report 10-shot accuracy (%, mean ± population std over three seeds); the last three columns report the change in the equal-weight four-dataset average relative to Full (pp) at each label budget, with the current budget highlighted. All retrained variants reuse the per-split hyperparameters of Full.

	10-shot accuracy					Δ Avg. vs. Full		
Variant	Cora	PubMed	Arxiv	Products	Avg.	3-shot	5-shot	**10-shot**
Full (ours)	**81.27**±1.36	**91.63**±0.83	**61.60**±0.86	**72.80**±0.82	**76.83**	–	–	–
Routing								
Confidence router	77.47±3.18	91.13±1.39	60.13±0.83	72.60±0.78	75.33	−2.73	−2.43	−1.49
Supervision								
w/o disagreement pseudo-labels	78.77±2.89	90.67±1.51	59.47±0.87	71.63±1.05	75.13	−2.94	−2.61	−1.69
Gold supervision only	78.77±2.89	90.67±1.51	59.47±0.87	71.63±1.05	75.13	−3.18	−2.31	−1.69
Representation								
G representation only	81.20±1.35	89.47±1.09	58.77±1.16	71.10±0.73	75.13	−2.78	−2.81	−1.69
L representation only	80.77±3.00	91.17±1.01	60.60±0.91	72.00±0.92	76.13	−1.25	−0.91	−0.69
Output								
Repair-only training	80.07±0.47	90.60±0.70	59.83±0.46	69.63±1.76	75.03	−2.16	−3.23	−1.79
w/o repair auxiliaries	**81.27**±1.36	**91.63**±0.83	59.70±2.22	71.37±0.49	75.99	−1.16	−1.37	−0.83
Reference (no retraining of Full)								
Equal expert fusion	73.87±1.20	89.93±1.10	54.53±3.67	70.47±1.41	72.20	−4.71	−4.32	−4.62
Validation-tuned expert fusion	81.00±0.99	90.20±1.36	55.87±2.71	70.50±1.36	74.39	−2.24	−2.33	−2.43
Residual readout	79.20±0.62	91.27±0.46	54.60±2.12	67.20±1.00	73.07	−3.98	−5.72	−3.76

H Additional Hyperparameter Sensitivity

Setup. Figure 7 in the main text varies the disagreement acceptance ratio ρ_D, the auxiliary residual coefficients, and the learning rate. Here we study the disagreement weight λ_D of Eq. 60, which scales both disagreement terms, and its interaction with ρ_D. Together, the two parameters determine how much supervision the residual classifier receives from routed disagreement pseudo-labels: ρ_D sets how many are accepted, and λ_D sets how strongly they are weighted. The protocol follows Figure 7: under the 5-shot setting, each run changes λ_D, or ρ_D and λ_D jointly, in the tuned configuration of a split, retrains the residual classifier, selects its checkpoint on the validation set, and reports the change in test accuracy relative to the tuned configuration, averaged over three seeds. Setting $\lambda_D = 0$ removes the disagreement supervision and corresponds to the ablation variant *w/o disagreement pseudo-labels* in Appendix G; this point is not repeated here.



(a) Disagreement weight λ_D

(b) Interaction between the acceptance ratio ρ_D and the weight λ_D

Figure 11: Sensitivity to the supervision from disagreement pseudo-labels (5-shot): paired accuracy change (pp) with respect to the tuned configuration, averaged over three seeds. (a) Disagreement weight λ_D (log scale; bars: ± 1 SD; shaded band: ± 1 pp). (b) Joint variation of the acceptance ratio ρ_D and the weight λ_D; colors are normalized within each dataset (darker is higher).

Disagreement weight. Any $\lambda_D \in [0.5, 2]$ keeps accuracy within 1.0 pp of the tuned configuration on every dataset, and the change averaged over datasets lies between -0.20 and -0.02 pp (Figure 11(a)). Smaller weights mainly hurt Products (-1.77 and -2.23 pp at $\lambda_D = 0.25$ and 0.1). Together with the ablation, this shows that routed disagreement pseudo-labels need a non-negligible weight to be useful, whereas their exact weight within a four-fold range matters little.

Interaction between acceptance and weight. Figure 11(b) shows a broad region of good configurations: 30 of the 45 combinations stay within 1 pp of the tuned configuration, all nine do so on PubMed, and eight of the nine improve on it on Citeseer. The preferred region, however, differs across datasets. On Cora, accepting more disagreement nodes is beneficial: every combination with $\rho_D \geq 0.5$ lies between -0.13 and $+0.83$ pp, whereas $\rho_D = 0.1$ leaves useful supervision unused. On Arxiv, a small weight works best ($+0.13$ to $+0.77$ pp at $\lambda_D = 0.25$), whereas on Products the best combinations pair a small acceptance ratio with a larger weight ($\rho_D = 0.1$, $\lambda_D \geq 1$), in line with the validation-selected ratio of 7.5% on all three Products splits (Table 7). Hence the number of routed disagreement pseudo-labels and their weight are complementary rather than interchangeable, and no single global combination suits every graph, which motivates selecting ρ_D and λ_D per split.

I GENERALIZATION ACROSS LLM BACKBONES

We keep all components of FairRouter unchanged and replace the LLM expert with Qwen3-8B (Yang et al., 2025) or Mistral-7B (Jiang et al., 2023). Table 2 compares FairRouter with direct inference of the corresponding frozen LLM at 10 shots, and Table 13 reports all label budgets.

FairRouter consistently improves its LLM expert. FairRouter outperforms direct LLM inference in 44 of the 45 backbone–dataset–budget settings, with average gains of 8.23, 17.06, and 7.35 pp at 10 shots for Qwen3-8B, Mistral-7B, and LLaMA3-8B, respectively. The single exception, Citeseer with LLaMA3-8B at 5 shots (-0.17 pp), occurs where the LLM alone is already stronger than every baseline in Table 1.

Gains are largest where the LLM is weakest. Across the 15 backbone–dataset pairs at 10 shots, the gain of FairRouter is strongly negatively correlated with the accuracy of direct LLM inference (Pearson $r = -0.73$, $p = 0.002$). The most striking case is Mistral-7B on Arxiv, where direct inference reaches only 25.23% and FairRouter reaches 54.33% (+29.10 pp). Conversely, Qwen3-8B is the strongest backbone on Arxiv (59.30%), and the gain there is the smallest (+3.50 pp). This inverse relation is what node-wise routing is designed to produce: when the semantic modality is unreliable, the router shifts trust toward the structural modality, and vice versa.

FairRouter is largely insensitive to the backbone. The average accuracy of direct inference varies by 13.15 pp across backbones (55.65–68.80%), whereas that of FairRouter varies by only 3.44 pp (72.71–76.15%). Across budgets, FairRouter's average accuracy increases monotonically from 3 to 10 shots for all three backbones, and even at 3 shots it surpasses direct inference by 5.99–13.75 pp on average.

J CASE STUDIES

Figure 6 examines individual nodes to illustrate how FairRouter makes its decisions.

R1: correcting the supervision direction. For a Products node whose ground truth is *Beauty*, the GNN predicts the semantically adjacent class *All Beauty*, and the LLM predicts the correct class. A GNN-preferring judge accepts the wrong GNN label because its margin (0.707) exceeds the threshold. FairRouter instead estimates that the LLM is the more likely correct modality (0.425 versus 0.219), ranks the node well within the budget, and supervises it with the correct LLM label. Distinguishing such fine-grained categories requires the product description rather than the co-purchasing structure, which is exactly the node-level information the router exploits.

R2: filtering unreliable supervision. For another Products node, the GNN and LLM predict *Office Products* and *Camera & Photo*, whereas the ground truth is *Electronics*. The judge accepts the GNN label with a high margin (0.836). FairRouter assigns the highest probability to the *neither-correct* state (0.536), so the node's selection score is low, and it is rejected (rank 35,604 beyond the budget of 2,812). This shows the value of modeling trust separately from preference: a confident preference between two wrong candidates does not produce a pseudo-label.

F1: preserving a correct semantic answer. For a Cora paper on measurement error models (ground truth *Probabilistic Methods*), both the zero-shot and few-shot fine-tuned LLMs are correct, but the LLM retrained by GNN-as-Judge predicts *Theory*. Fine-tuning with GNN-preferred pairs can thus overwrite semantic knowledge that the LLM already had. FairRouter keeps both models frozen and fuses them, so every component predicts the correct class, and the final prediction assigns it a probability of 0.78.

F2: repairing beyond both models. For a Cora paper on constructive induction (ground truth *Theory*), both models and their mixture predict *Rule Learning*, and the LLM assigns the true class a probability of almost zero. The residual branch nevertheless places its mass on *Theory*, and the fused prediction recovers the correct class (0.53). This is the scenario in Challenge 2: routing alone cannot help when both candidates are wrong, whereas the cross-modal residual can propose a class beyond either expert's top-1 prediction.

Table 13: Accuracy (%) of FairRouter under 3-, 5- and 10-shot settings with different LLM backbones. LLM Inference does not depend on the label budget. The best result in each row is in **bold**.

Dataset	LLM Inference	FairRouter		
		3-shot	5-shot	10-shot
Qwen3-8B				
Cora	61.80±1.68	78.60±0.75	79.30±0.91	**81.50**±1.44
Citeseer	63.00±0.57	68.73±1.25	71.53±1.88	**72.87**±1.24
PubMed	85.70±1.10	88.70±1.22	89.67±0.76	**89.80**±0.86
Arxiv	59.30±0.54	**64.17**±2.69	62.83±1.68	62.80±0.70
Products	65.70±2.08	67.73±1.11	69.57±2.35	**69.67**±1.45
Avg.	67.10	73.59	74.58	**75.33**
Mistral-7B				
Cora	63.70±1.93	79.57±1.52	78.83±1.91	**80.57**±2.44
Citeseer	52.67±0.80	69.77±2.37	70.57±1.21	**71.13**±2.21
PubMed	83.50±0.57	88.10±0.62	89.83±0.62	**90.50**±0.73
Arxiv	25.23±0.91	47.17±2.44	52.87±0.90	**54.33**±0.82
Products	53.13±1.53	62.40±2.82	59.80±2.19	**67.00**±2.26
Avg.	55.65	69.40	70.38	**72.71**
LLaMA3-8B				
Cora	66.53±1.47	79.37±0.17	80.70±1.91	**81.27**±1.36
Citeseer	72.77±0.48	73.37±0.12	72.60±0.79	**73.47**±1.25
PubMed	84.47±1.25	90.80±1.04	91.47±1.30	**91.63**±0.83
Arxiv	51.73±0.58	58.13±0.12	60.40±1.56	**61.60**±0.86
Products	68.50±0.33	72.30±0.85	72.43±0.41	**72.80**±0.82
Avg.	68.80	74.79	75.52	**76.15**

K Limitations and Future Work

Frozen models bound the attainable accuracy. FairRouter never updates the GNN or the LLM, which keeps it lightweight but limits its accuracy when both models are weak on a fine-grained label space. Products is the example: FairRouter produces the most precise pseudo-labels there, yet it trails GNN-as-Judge, which fine-tunes the LLM. Since routing is agnostic to the downstream learner, feeding routed pseudo-labels into LLM or GNN fine-tuning is a natural extension.

Dataset-specific router configurations. The scorer family of the router is selected per dataset and its acceptance ratios per split, both on the validation set (Appendices B and E.2). Although the *Confidence router* ablation confirms that the gains do not stem from the number of accepted nodes, a single configuration shared across datasets would further simplify deployment.

Combining learned routing with confidence priors. On graphs dominated by one modality, such as PubMed and Products, expert confidence is a strong disagreement signal that the router does not fully recover from few labels (Figure 10). Incorporating such priors into the router is a promising direction.

Beyond two modalities. FairRouter routes between one GNN and one LLM. Extending the trust and preference factorization to more than two models, such as several LLMs or multimodal node attributes, is left for future work.