MIDAS: MULTI-TEACHER KNOWLEDGE DISTILLATION WITH ADAPTIVE STRUCTURE-AWARE EMBEDDINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Graph Neural Networks (GNN) achieve strong performance on graph-structured data but incur high inference latency due to iterative neighborhood aggregation. To address this, knowledge distillation helps translate a heavy GNN into a lightweight multi-layer perceptron. However, these methods are constrained by the choice of a single teacher, which can be sub-optimal in structurally diverse regions. We propose MIDAS, a multi-teacher distillation framework designed to obtain optimal supervision from GNN architectures across structurally diverse regions. The student MLP thus benefits from the locally optimal teacher rather than a globally best one. MIDAS realizes this through: (1) a dual-encoder contrastive alignment module that embeds structural identity into node representations without graph access at inference, (2) a density-based clustering and cluster-conditional teacher routing mechanism that assigns each node to its locally optimal teacher, and (3) a trunk-adapter student architecture with a two-phase training strategy that prevents gradient interference across divergent teacher signals. Experiments on seven benchmark datasets show that MIDAS outperforms 17 baselines across single-teacher and ensemble distillation settings, achieving up to 20$\times$ inference speedup, without sacrificing generalization

1 INTRODUCTION

Graph Neural Networks (GNNs) Kipf & Welling (2016) have revolutionized relational representation learning, achieving state-of-the-art performance in applications such as finance Cheng et al. (2025), social networks Sharma et al. (2024), molecular property prediction Kang et al. (2022); Liu et al. (2023), and recommendations He et al. (2020). Despite their empirical success, deploying GNNs in real-world production environments remains a challenge. Iterative neighborhood aggregation in GNN architectures introduces significant computational overhead Zhang et al. (2021a) . For large-scale graphs containing millions to billions of nodes, the exponential growth in neighborhood aggregation computations makes GNNs impractical for latency critical tasks. Most GNNs also require explicit graph structure at inference, increasing memory overhead. Although methods such as pruning Ding et al. (2021), partitioning Modak et al. (2024), and quantization Zhou et al. (2021) reduce the latency of GNNs, the core dependency on neighborhood aggregation remains unresolved.

To bridge this gap, knowledge distillation Hinton et al. (2015) into multi-layer perceptrons (MLPs) emerged as a principled solution. A lightweight graph-free student is trained to mimic a complex GNN teacher, recovering most of the predictive performance without requiring an adjacency matrix at inference. Several works have explored this direction Lu et al. (2024); Tian et al. (2022); Wu et al. (2023); Yang et al. (2023); Zhang et al. (2021a), incorporating structural knowledge through positional embeddings, structural codebooks, node reliability estimation, and related mechanisms.

However, we argue that single-teacher distillation is inherently suboptimal. Different GNN architectures like GCN, GAT, and GRAPHSAGE employ distinct aggregation mechanisms and therefore capture fundamentally different structural cues. Pei et al. Pei et al. (2020) showed that GCN and GAT performance varies systematically with local homophily and node degree, suggesting that GNN performance is *non-uniformly distributed* across structurally distinct graph regions. This is further corroborated by Tables 2 and 3 of this paper, which show that no single teacher dominates across

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.

all node types. Thus, no single teacher can be expected to excel uniformly across all nodes, where neighborhoods may exhibit substantially different structural characteristics Morris et al. (2019); Xu et al. (2018). This motivates routing supervision from the locally best-matched teacher for each structural region, rather than distilling from a single globally fixed teacher. This localized supervision can also improve calibration, as each region can learn from the most reliable teacher, instead of overfitting on any specific teacher

Realizing this idea requires three capabilities: (i) a mechanism to identify structural regions without graph access at inference; (ii) a routing strategy that assigns nodes to their best-matched teacher; and (iii) a training procedure that prevents gradient interference when multiple conflicting teacher signals are active.

In this work, we propose MIDAS (Multi-Teacher Knowledge Distillation with Adaptive Structure-Aware Embeddings), a multi-teacher graph-to-MLP distillation framework for graph-free inference. Our contributions are:

- **Region-based teacher routing.** We cluster the node into a structure-aware feature space and assign each region the teacher GNN that best matches its local structural context, yielding heterogeneous supervision tailored to the inductive biases in each region.
- **Structure-aware feature embeddings.** To ensure clusters can be discoverable from node features alone during inference, we introduce a dual-encoder contrastive training module to align node features with the structural embeddings in a shared latent space. After alignment, these structural regions become discoverable directly from node features, thus enabling both cluster formation and graph-free routing at inference.
- **Trunk-adapter student with two-phase training.** A shared trunk processes the full representation space, while lightweight cluster-specific adapters specialize for each structural region. We first warm up the trunk under the globally best teacher, then freezes the trunk and trains adapters under their assigned teachers, mitigating gradient interference across divergent teachers signals Yu et al. (2020); Shi et al. (2023); Liu et al. (2021).

To evaluate MIDAS, we conduct extensive experiments on seven benchmark datasets, both homophilic and heterophilic. Our results demonstrate that our method MIDAS outperforms 17 baselines while achieving superior prediction calibration, topological alignment and up to 20x inference speedup over teacher GNNs. Theoretically, we demonstrate the optimality of the two phase training strategy and that the misrouting rate of feature-only routing is controlled by the alignment loss together with the separation of the structural regions

2 Related Work

Graph representation learning encodes structure via random walks (DeepWalk Perozzi et al. (2014), node2vec Grover & Leskovec (2016)), multi-hop patterns (Struc2Vec Ribeiro et al. (2017)), or contrastive alignment (DGI Veličković et al. (2018), MVGRL Hassani & Khasahmadi (2020), GRACE Zhu et al. (2020b), GraphCL You et al. (2020)). Inductive methods Chen et al. (2022); Li et al. (2020); Teru et al. (2020); You et al. (2021) also inject structure into message passing. However, these approaches couple structural signals with graph-dependent computation, rendering them unsuitable for GNN-to-MLP distillation where graphs are unavailable at inference.

2.1 Distilling GNNs into MLP

Recent work has explored distilling GNNs into graph-free MLP students Zhang et al. (2021a); Tian et al. (2022). GLNN Zhang et al. (2021a) pioneered this direction by training an MLP to mimic a pre-trained GNN teacher through KL divergence minimization, enabling graph-free inference. NOSMOG Tian et al. (2022) augments the student with positional embeddings derived from random walks to explicitly incorporate structural knowledge. VQGraph Yang et al. (2023) learns a structural codebook via vector quantization during GNN pre-training before distilling into an MLP. KRD Wu et al. (2023) improves distillation reliability by quantifying teacher confidence and down-weighting uncertain supervision signals. AdaGMLP Lu et al. (2024) uses AdaBoosting over an ensemble of MLP students to improve robustness.SALE-MLP Pal et al. (2025), employs unsupervised structural losses for distilling structural knowledge into an MLP

Despite these advances, most existing GNN-to-MLP distillation methods rely on a single GNN teacher, implicitly assuming that one architecture provides uniformly reliable supervision for all nodes. As per Pei et al. (2020), the best teacher is thus unlikely to be globally fixed. Effective distillation should instead account for teacher specialization across structurally distinct regions Xu et al. (2018); Morris et al. (2019).

2.2 MULTI-TEACHER DISTILLATION

Multi-teacher distillation in vision relies on sample weighting Zhang et al. (2022b); Yang et al. (2025) or gradient conflict resolution Du et al. (2020), ignoring graph topologies. GNN-to-GNN multi-teacher methods aggregate complementary structural knowledge Zhang et al. (2021b; 2022a) or logits Wang et al. (2021); Yun et al. (2022), but still require graph access during inference.

For GNN-to-MLP scenarios, Yang et al. Yang et al. (2024) build a combined "super-teacher," while Chao et al. Chao et al. (2025) use contrastive transfer. Both aggregate teacher signals globally, ignoring local structural context. This global aggregation creates two major challenges: it obscures localized teacher expertise (e.g., performance on dense vs. sparse neighborhoods), and it forces a shared student to optimize diverging teacher signals. This introduces gradient conflicts Liu et al. (2021); Shi et al. (2023); Yu et al. (2020) that can degrade performance below even single-teacher baselines Shin & Shin (2023).

MIDAS addresses these gaps by aligning feature and structure embeddings, clustering them into regions routed to their best teachers and training teacher-specific adapters over a shared trunk.

3 PRELIMINARIES AND PROBLEM FORMULATION

A graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ has node set $\mathcal{V}$, edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and features $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ with rows $\mathbf{x}_v$. Nodes are split into $\mathcal{V}_{\text{train}}$, $\mathcal{V}_{\text{val}}$, and $\mathcal{V}_{\text{test}}$; training and validation nodes carry a class $y_v \in \mathcal{Y}$. A GNN embeds nodes as $\mathbf{Z} \in \mathbb{R}^{|\mathcal{V}| \times c}$. We are given K pre-trained teachers $\{\Theta_t^1, \ldots, \Theta_t^K\}$, and let their predictive distributions be. $p_{\Theta_t^k}(\cdot \mid v)$. Let $\psi : \mathcal{V} \to \mathbb{R}^{c_s}$ be the structural descriptor derived from walk patterns. In MIDAS setting, the model's prediction for v depends on $\mathbf{x}_v$ alone.

Problem 1 (Structure-free structural partitioning). Learn an encoder f_{feat} and a partition $\pi : \mathcal{V}_{\text{train}} \cup \mathcal{V}_{\text{val}} \to \{-1, 1, \ldots, L\}$ such that nodes sharing a region have similar ψ, along with a mapping $\widehat{\pi} : \mathcal{V} \to \{1, \ldots, L\}$ that is computable from $\mathbf{x}_v$ alone and extends π to unseen nodes.

$$\mathbf{z}_v^{\text{feat}} = f_{\text{feat}}(\mathbf{x}_v) \;\text{ fitted against }\; \psi(v), \tag{1}$$

We can leverage structural information to fit f_{feat}. The partition recoverable at inference is an approximation to one defined on ψ directly.

Problem 2 (Region-routed distillation). Given the partition, let $\kappa(v) \in \{1, \ldots, K\}$ denote the teacher assigned to the region of v. Distill a structure-free student so that each node is supervised by its region's teacher:

$$\min_{f_s \in \mathcal{F}} \sum_{v \in \mathcal{V}_{\text{train}}} \mathcal{L}\Big(p_{\Theta_t^{\kappa(v)}}(\cdot \mid v),\; f_s(\mathbf{x}_v)\Big), \tag{2}$$

where $\mathcal{F}$ is the class of structure-free models and $\mathcal{L}$ a distillation loss. Solving Problem 1 helps us learn where each node belongs structurally without graph access at inference, and Problem 2 relies on these partitions for optimal teacher assignment

4 THE MIDAS FRAMEWORK

MIDAS distills K teacher GNNs into a single structure-free student (Figure 1). The key is to build a partition into structural regions in a space that remains accessible without the graph. MIDAS aligns a feature encoder with a structural encoder, to predict structural context from node features. In this aligned space, it partitions the training and validation nodes into regions, assigning the best teacher and places unseen nodes by nearest region center. Finally, a shared trunk with lightweight adapter per teacher is distilled so that every node is supervised by its region's teacher.

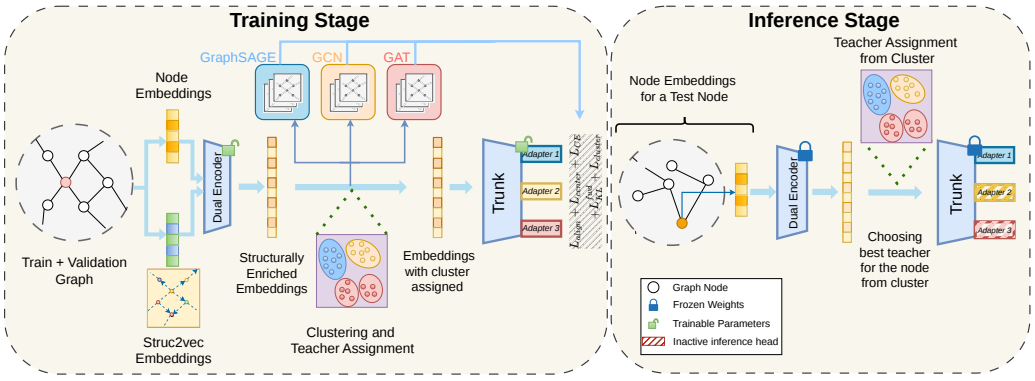


Figure 1: Overview of MIDAS. During inference, MIDAS does not use the graph

4.1 Learning Structure-Aware Feature Representations

During training each node has two views: its features $\mathbf{x}_v$ and a structural descriptor $\psi(v)$ computed from the graph. At inference only the features remain.We therefore train a feature encoder f_{feat} that, given $\mathbf{x}_v$ alone, places nodes with similar structural context close together, so that the resulting space can be partitioned into regions with well-defined centers and an unseen node can be placed by distance to those centers (Sec. 4.2). **Structural descriptor and alignment.** We compute a structural descriptor $\psi(v)$ using Skip-Gram (`word2vec`) on degree-aware random walks. Unlike Struc2Vec Ribeiro et al. (2017), which builds a separate multilayer context graph to link distant nodes, our walks remain on the input graph to capture local, degree-aware neighborhoods. From a current node w, the next node $u \in \mathcal{N}(w)$ is sampled with probability $P(v_t = u \mid v_{t-1} = w) \propto \exp\big(-|\deg(w) - \deg(u)|/\sigma\big)$, where σ is the mean degree.

To align the structural and feature views without labels, we use mappings $f_{\text{feat}} : \mathbb{R}^d \to \mathbb{R}^{c'}$ and $f_{\text{struct}} : \mathbb{R}^{c_s} \to \mathbb{R}^{c'}$ to generate ℓ_2-normalized embeddings $\mathbf{z}_v^{\text{feat}}$ and $\mathbf{z}_v^{\text{struct}}$. These are trained via a unidirectional InfoNCE loss Oord et al. (2018) over batches of N nodes:

$$\mathcal{L}_{\text{align}} = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{\exp\big(\text{sim}(\mathbf{z}_{v_i}^{\text{feat}}, \mathbf{z}_{v_i}^{\text{struct}})/\tau\big)}{\sum_{j=1}^{N} \exp\big(\text{sim}(\mathbf{z}_{v_i}^{\text{feat}}, \mathbf{z}_{v_j}^{\text{struct}})/\tau\big)}, \tag{3}$$

where sim is cosine similarity and $\tau > 0$. The feature embedding allows f_{struct} to teach f_{feat} the aspects of ψ that are predictable from node features $\mathbf{x}_v$. (App. A.8).

Partitioning and tightening the aligned space. While alignment brings structurally similar nodes closer, we partition the space once with DBSCAN Ester et al. (1996), which does not have a predefined number of groups, to uncover regions where teachers naturally excel.

$$\pi = \text{DBSCAN}\big(\{\mathbf{z}_v^{\text{feat}}\}_{v \in \mathcal{V}_{\text{train}} \cup \mathcal{V}_{\text{val}}}, \epsilon, m\big), \qquad \pi(v) \in \{1, \ldots, L\} \cup \{-1\}, \tag{4}$$

where $\pi(v) = l$ assigns v to region l, $\pi(v) = -1$ marks noise, and L is the number of regions returned. Each region $c_l = \{v : \pi(v) = l\}$ has center $\mathbf{C}_l = \text{normalize}\big(\frac{1}{|c_l|}\sum_{v \in c_l} \mathbf{z}_v^{\text{feat}}\big)$, which determines how training nodes are supervised and maps unseen nodes at inference.

Regions bound by density might be elognated, making assignment based on center nodes unreliable, and hence we add a center loss L_{center} Wen et al. (2016) over the clustered nodes for better alignment. The loss pulls nodes towards their own center, without pushing regions outward. In the final step, we add a cosine classifier with ℓ_2-normalized class vectors $\{\mathbf{w}_y\}_{y \in \mathcal{Y}}$ to make embeddings for unseend nodes less sensitive to task-irrelevant feature variation, ensuring test embeddings land near the centre more reliably.The feature encoder f_{feat} is then frozen for inference after discarding f_{struct} and the class vectors, ensuring any node can now be embedded to the shared latent space. without access to the graph. Further theoretical details of this section is in A.8

4.2 Selecting Teachers and Routing Nodes

Sec. 4.1 produced a frozen feature encoder f_{feat}, a partition π of the training and validation nodes into L regions, and one stored center $\mathbf{C}_l$ per region. We first assign each sufficiently large region to the most optimal teacher and finally distill the student in two phases: a warm-up in which a shared trunk learns from all teachers, and a specialization in which one adapter per teacher learns from that teacher alone.

Assigning teachers to regions. Let $\mathcal{R} \subseteq \{1, \dots, L\}$ be the regions with at least $n_{\text{min}} = 25$ training nodes and let $\mathrm{Acc}(\Theta, \mathcal{S})$ for the accuracy of a teacher Θ on a node set $\mathcal{S}$; smaller regions and DBSCAN noise ($\pi(v) = -1$) contain too few labeled nodes to compare teachers reliably and take no part in selection. Each $l \in \mathcal{R}$ is assigned the teacher most accurate on its training nodes,

$$k_l^* = \underset{k \in \{1,\dots,K\}}{\operatorname{argmax}} \ \mathrm{Acc}\big(\Theta_t^k,\ \mathcal{V}_{\text{train}} \cap c_l\big)\,, \qquad l \in \mathcal{R}. \tag{5}$$

Validation nodes take part in forming the regions (Eq. 4) but not in comparing teachers, so the selection uses no held-out labels. App. A.6 gives the corresponding risk bound. Every training node is then mapped to the optimal teacher based on its region. Training nodes with $\pi(v) = -1$ or $\pi(v) \notin \mathcal{R}$ belong to no retained region. They are supervised by the globally most accurate teacher. For test nodes, the stored center and the frozen encoder maps these nodes into the region based on the nearest center.

Student architecture. The student pairs a shared trunk g_θ with one adapter per teacher in use, $\{a_{\varphi_k}\}_{k \in \mathcal{K}}$, whose hidden width is 0.1–$0.3\times$ that of the trunk. Adapters are indexed by teachers. Trunk and adapters are MLPs, so f_s is structure-free. Distillation runs for E_{s} epochs under a single objective whose supervision changes at epoch $E_{\text{sw}} < E_{\text{s}}$:$f_s(\mathbf{x}_v) = a_{\varphi_{\kappa(v)}}(g_\theta(\mathbf{x}_v))$.

$$\mathcal{L}(e) = \gamma_{\text{ce}}\,\mathcal{L}_{\text{CE}} + \frac{\gamma_{\text{kl}}}{|\mathcal{V}_{\text{train}}|} \sum_{v \in \mathcal{V}_{\text{train}}} \begin{cases} \dfrac{1}{K} \displaystyle\sum_{k=1}^{K} \mathcal{L}_{\text{KL}}\big(v, \Theta_t^k\big)\,, & e \le E_{\text{sw}} \quad \text{(warm-up)}, \\ \mathcal{L}_{\text{KL}}\Big(v, \Theta_t^{\kappa(v)}\Big)\,, & e > E_{\text{sw}} \quad \text{(specialization)}, \end{cases} \tag{6}$$

where $\mathcal{L}_{\text{CE}}$ is cross-entropy of p_s against the labels on $\mathcal{V}_{\text{train}}$, and $\mathcal{L}_{\text{KL}}(v, \Theta) = T_{\text{KD}}^2\,\mathrm{KL}(p_\Theta(\cdot \mid v) \,\|\, p_s(\cdot \mid v))$. During warm-up, the shared trunk captures representations useful across regions, learning from all K teachers. During specialization, this head is discarded and the adapter k therefore sees only nodes assigned to teacher k and only teacher k's distributions; with $\partial\mathcal{L}/\partial\theta = 0$ and the adapters holding disjoint parameters, divergent teacher signals never conflict over weights. The importance of this two phase training is given in A.9. During inference, the test node is encoded once by f_{feat}, and passed through the trunk and adapter, without any access to the graph. For a detailed explanation of the training procedure and complexity, please refer to A.3 and A.2.

5 Experiments

We evaluate MIDAS on the following research questions. **RQ1**: Comparative Performance(Transductive and Inductive) **RQ2**: Structural Specialization and Gradient Inteference Analysis **RQ3**: Component Analysis **RQ4**: Structure-Aware Embedding Quality **RQ5**: Calibration Analysis **RQ6**: Inference Latency

5.1 Experimental Setup

Datasets. Table 8 in A.1 summarizes the seven datasets. Actor Pei et al. (2020), Flickr Zeng et al., Amazon-Ratings Platonov et al., and Questions Platonov et al. are heterophilic. Questions has high edge homophily because 97% of its nodes share one class, and is heterophilic under the adjusted measure of Platonov et al. Platonov et al.. Our selection of these datasets intentionally departs from the narrow focus on CPF Sen et al. (2008) and Amazon family Shchur et al. (2018) that dominate prior work Zhang et al. (2021a); Tian et al. (2022); Wu et al. (2023) to enable an accurate representation of model performance over a range of degree profiles, homophily levels and domain specific patterns. Each set of experiments is run 10 times with different random seeds and

Table 1: Inductive setting: accuracy (%), $\mu \pm \sigma$ over 10 seeds; ROC-AUC (%) for Questions

Teacher	Method	Actor	PubMed	A-Photo	Co-CS	Flickr	Questions*	A-Ratings
GCN	Standard	27.70±0.41	78.80±0.54	90.70±0.53	90.60±0.46	49.20±0.44	64.17±1.52	47.50±0.34
	GLNN	30.17±0.44	76.07±0.33	89.02±0.51	84.78±0.36	48.11±0.29	68.07±1.43	47.54±0.21
	NOSMOG	29.64±0.39	76.66±0.41	89.38±0.20	86.03±0.45	48.34±0.22	68.96±1.22	47.54±0.21
	KRD	29.29±0.32	77.05±0.38	89.62±0.30	86.86±0.37	48.50±0.39	68.06±0.45	47.53±0.51
	AdaGMLP	29.02±0.40	77.34±0.29	89.80±0.37	87.48±0.34	48.62±0.23	70.23±1.43	47.52±0.32
	VQGraph	28.67±0.34	77.73±0.50	90.04±0.22	88.32±0.43	48.77±0.43	72.51±0.67	47.52±0.45
	SALE-MLP	34.39±1.05	76.61±2.37	89.63±1.75	88.23±3.29	48.84±0.33	71.03±0.81	47.72±0.14
GAT	Standard	28.90±0.48	79.10±0.24	91.60±0.27	90.90±0.39	49.50±0.25	71.05±1.26	47.80±0.36
	GLNN	31.03±0.51	76.29±0.23	89.67±0.48	85.00±0.50	48.32±0.23	69.71±1.43	47.76±0.39
	NOSMOG	30.57±0.21	76.89±0.23	90.08±0.46	86.26±0.28	48.58±0.34	74.79±0.37	47.77±0.50
	KRD	30.27±0.52	77.29±0.26	90.36±0.20	87.11±0.34	48.74±0.52	73.85±0.47	47.77±0.30
	AdaGMLP	30.04±0.44	77.59±0.46	90.56±0.47	87.74±0.43	48.87±0.37	72.89±0.27	47.78±0.28
	VQGraph	29.74±0.22	78.00±0.46	90.84±0.22	88.58±0.31	49.04±0.22	73.95±0.37	47.78±0.52
	SALE-MLP	34.21±0.66	76.26±1.12	89.62±1.84	88.67±2.17	48.83±0.33	74.02±0.98	47.28±0.20
GRAPHSAGE	Standard	32.80±0.39	80.00±0.22	92.30±0.38	91.70±0.50	50.30±0.22	70.70±0.22	48.40±0.50
	GLNN	33.84±0.34	76.94±0.53	90.17±0.40	85.58±0.40	48.90±0.21	75.14±0.23	48.19±0.43
	NOSMOG	33.62±0.40	77.59±0.31	90.63±0.29	85.19±0.43	49.20±0.31	74.26±0.29	48.24±0.25
	KRD	33.47±0.43	78.03±0.36	90.93±0.53	87.76±0.53	49.40±0.20	75.34±0.42	48.27±0.40
	AdaGMLP	33.36±0.23	78.36±0.39	91.16±0.38	88.42±0.25	49.55±0.32	76.40±0.22	48.29±0.29
	VQGraph	33.21±0.30	78.80±0.35	91.46±0.39	89.29±0.31	49.75±0.54	75.48±0.48	48.32±0.39
	SALE-MLP	34.34±1.48	79.06±2.14	89.19±0.56	88.09±2.98	49.95±0.22	75.02±1.09	47.60±0.37
–	MLP	36.91±0.52	69.36±0.44	85.20±0.38	70.23±0.61	46.20±0.35	70.80±0.29	47.96±0.33
All three	Mean	34.80±0.40	79.80±0.37	92.30±0.48	92.00±0.23	50.90±0.46	75.70±0.28	48.20±0.47
	Confidence	35.70±0.42	78.20±0.25	92.50±0.28	92.30±0.26	51.10±0.30	75.91±0.41	48.40±0.46
	Two-heads	36.40±0.35	79.60±0.28	92.70±0.47	92.60±0.22	51.30±0.39	74.10±0.45	48.60±0.21
	Soft MoE	36.90±0.30	79.80±0.24	92.90±0.49	92.80±0.25	51.60±0.33	75.20±0.41	48.80±0.37
	AE-KD	35.26±0.95	78.16±1.47	90.06±2.88	93.01±0.30	49.15±1.46	76.17±1.07	48.11±0.27
	CA-MKD	34.52±0.41	78.63±1.30	90.91±3.02	93.05±0.31	50.75±0.61	77.56±0.12	47.51±0.67
	MTKD-RL	34.85±0.76	78.58±1.66	91.24±2.22	92.96±0.32	49.03±0.43	74.13±1.17	48.12±0.56
MIDAS (ours)		**38.75**†±0.38	**81.70**†±0.35	**93.01**±0.27	**93.10**±0.29	**51.85**±0.41	**79.09**±0.37	**49.46**†±0.31

average accuracy is reported except for Questions, following Platonov et al., which is evaluated on ROC-AUC as accuracy on a 97%-majority binary task is uninformative

Teachers and baselines We adopt three backbone GNN architectures: GAT, GRAPHSAGE, and GCN; which serve both as individual baselines and as the collective teacher ensemble for the MIDAS framework. Additionally, we use these three GNN architectures as underlying teacher models for established single-GNN-to-MLP benchmarks, including GLNN Zhang et al. (2021a), NOSMOG Tian et al. (2022), KRD Wu et al. (2023), AdaGMLP Lu et al. (2024), VQGraph Yang et al. (2023) and SALE-MLP Pal et al. (2025). The only published multi-teacher GNN-to-MLP baseline is Yang et al. (2024)'s two-head student. In the absence of other established benchmarks, we introduce 3 baselines using the same teacher to further validate our framework. *Mean* and *Confidence* KL-distill one MLP per teacher, then average their class probabilities or select the most confident student's prediction. *Soft MoE* uses one student and a feature-only gate $g(\mathbf{x}_v) \in \Delta^{K-1}$. It distills the gate-weighted teacher mixture $\sum_k g_k(\mathbf{x}_v)\, p_t^{(k)}(\cdot \mid v)$ with MIDAS's cross-entropy weight, temperature, and KL direction (Eq. (6)), testing whether hard regional assignment is needed. Additionally, we further adapt 3 multi-teacher methods from vision to our multi teacher setting. .CA-MKD Zhang et al. (2022b), scores teachers by confidence against ground-truth.AE-KD Du et al. (2020) solves ensembles as a multi-objective optimization. MTKD-RL Yang et al. (2025), is the lateset state of the art architecture in the family of learning pre-instance teacher weights with a dedicated module and optimizes multi teacher weights by constructing teacher performance.

Implementation Details and Experimental Setting For transductive and inductive settings, we follow the protocol of Zhang et al. (2021a); Tian et al. (2022). For the datasets, we use the splits given in the respective papers. For all datasets, we use the public splits from the original papers Pei et al. (2020); Yang et al. (2016); Shchur et al. (2018); Zeng et al.; Platonov et al.. All analyses except RQ1 and RQ6 use the inductive setting, which more faithfully matches the real-world settings. All the hyperparameters, model dimensions, hardware settings are present in B.4,A.10 and A.11.

Table 2: Test accuracy (%) per assigned region and fraction of test nodes routed there in Actor

Region (assigned)	Nodes (%)	GCN	GAT	GRAPHSAGE
GRAPHSAGE (high deg. var.)	72.4	30.0	29.0	**36.4**
GCN (high PageRank)	14.6	**44.0**	34.0	31.0
GAT (high LCC, low deg.)	13.0	33.0	**46.0**	30.0

Table 3: Teacher utilization: training nodes assigned to each teacher (%)

	Actor	PubMed	Co-CS	A-Photo	Flickr	Questions	A-Ratings
GRAPHSAGE	72.4	58.2	25.3	31.2	44.1	38.9	51.3
GAT	13.1	41.8	32.5	55.6	28.7	35.4	22.6
GCN	14.6	0.0	42.2	13.3	27.2	25.7	26.1
			...				

5.2 COMPARATIVE PERFORMANCE(RQ1)

Tables 10 and 1 report node classification results across both homophilic and heterophilic datasets under transductive and inductive settings. Overall, MIDAS consistently outperforms all baselines, including strong GNN and MLP models, demonstrating its effectiveness even on heterophilic datasets. Hard teacher assignment outperforms soft teacher assignment as latter exposes nodes to mismatched teachers, degrading performance. While other ensemble methods like mean and confidence show modest improvements, naive aggregation ignores graph topology and applies a one-size-fits-all combination strategy, making adaptive teacher selection based on structural context more effective than uniform aggregation The generalizability of our method to unseen nodes is further evident from the inductive results, where it surpasses other methods by a large margin, particularly on the Actors dataset. On the heterophilic Actor and Amazon-Ratings the plain MLP beats most single-teacher students: aggregating dissimilar neighbors is harmful there, and a student that imitates one GNN everywhere inherits the damage. MIDAS still outperforms the MLP because its supervision is regional. The encoder is fitted to degree-aware walk statistics rather than edge-wise label smoothing, so regions reflect structural context, and a teacher misled by heterophilous edges in one part of the graph is not imposed on the rest. Further analysis on topological quality is in A.4

5.3 STRUCTURAL SPECIALIZATION AND GRADIENT INTERFERENCE (RQ2)

Tables 2 and 3 shows no single teacher wins everywhere. On Actor, GAT is most accurate in regions with high clustering coefficient and low degree, GCN in regions with high PageRank, and GRAPHSAGE in the degree-heterogeneous majority. These descriptors characterize the regions post hoc and they are not used to form them, and suggest that MIDAS recovers semantically coherent structure. Eq. (5), each region's teacher is selected by accuracy on its training nodes. Across all seven datasets, at least two teachers are selected ($|\mathcal{K}^*| \geq 2$), the globally most accurate teacher $k^\star$ wins at least one region. This confirms that routing does not collapse into a single teacher fallback.

To quantify the interference that specialization removes, we take the trunk θ at the final warm-up epoch and, for each node v and teacher pair (k, k'), compute the cosine similarity of the two KL gradients with respect to θ. The conflict of v is the worst pair is binned as High ($c < -0.5$), Mid ($-0.5 \leq c < 0$), Low ($0 \leq c < 0.5$), and None ($c \geq 0.5$). Table 4 bins *test* nodes: their gradients are evaluated post hoc through the frozen warm-up trunk using the teachers' inference-time distributions. MIDAS's gain over GRAPHSAGE grows with conflict: it is largest in the High and Mid bins and smallest where the teachers agree. Gains concentrated where teacher gradients oppose each other are direct evidence that interference on the shared trunk is what specialization removes. Specialization freezes the trunk and places each node with a disjoint adapter, so these gradients no longer meet on shared parameters.

5.4 COMPONENT ANALYSIS (RQ3)

Table 5 reports systematic ablations across four datasets to isolate the contribution of each MIDAS module.Omitting shared trunk warm-up degrades performance, as adapters must learn representations from scratch using only their assigned teacher's nodes rather than the full graph. w/o Specialization, updating shared parameters instead of using separate adapters for teacher-specific signals (Eq. (6)) causes substantial drops. A single shared classifier cannot adequately capture heterogeneous structural patterns. w/o dual encoder, applying DBSCAN directly to ℓ_2-normalized raw features $\mathbf{x}_v$ hurts performance most on Actor (where features lack structural context) and least on Coauthor-CS (where features are highly informative). This highlights the dual-encoder's necessity for generating structural proxies for unseen nodes. W/o Teachers, training with cross-entropy alone ($\gamma_{\text{kl}} = 0$ in Eq. (6)) retains the region-based adapters and still outperforms MLP on Actor and Coauthor-CS, showcasing the importance of region specific adapters. However, it trails the full

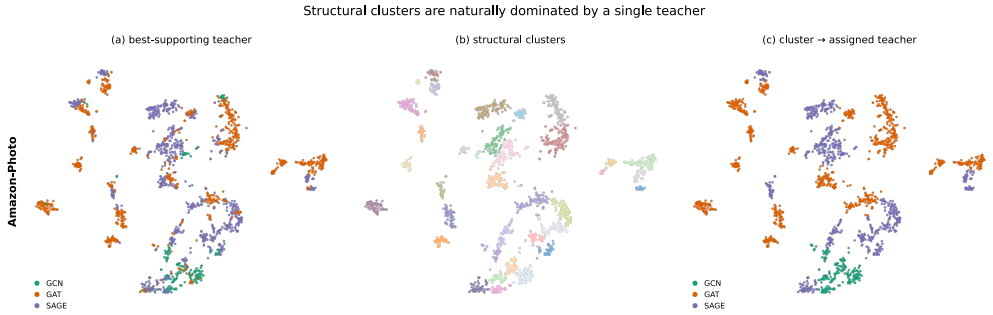


Figure 2: t-SNE of Amazon-Photo test nodes in $f_{\text{feat}}(\mathbf{x}_v)$. **Left:** Colored by true best teacher (ties broken by confidence; grey if none are correct). **Middle:** Colored by nearest-center region. **Right:** Regions colored by assigned teacher.

Table 4: Gradient conflict analysis (%) of MIDAS and GRAPHSAGE per conflict bin

Dataset	Conflict	Nodes (%)	MIDAS	GRAPHSAGE
Actor	High	3.3	95.8	85.0
	Mid	6.1	70.7	47.3
	Low	18.2	58.9	50.4
	None (agree)	72.4	28.4	24.8
PubMed	High	1.8	87.3	75.5
	Mid	3.6	84.2	73.3
	Low	5.2	83.3	76.7
	None (agree)	89.4	81.4	80.6

Table 5: Component analysis: inductive accuracy (%)

Configuration	Actor	PubMed	A-Ratings	Co-CS
MIDAS (full)	38.75	81.70	49.46	93.10
w/o warm-up	33.12	72.10	43.67	62.33
w/o specialization	35.76	77.40	47.23	86.01
w/o dual encoder	26.47	69.86	45.65	90.25
w/o teachers (γ_{kl}=0)	37.72	69.24	46.43	72.18
DBSCAN → K-means	36.10	78.80	46.90	90.40
MLP	36.91	69.36	47.96	70.23

model everywhere, confirming that regional teacher distillation drives the primary gains. Replacing DBSCAN with K-means reduces accuracy across all datasets. Nodes favoring a specific teacher form clusters that are neither convex nor uniformly sized, causing K-means Voronoi cells to improperly split or merge natural boundaries. Table 6 evaluates the structure-aware embedding's training strategy. Using alignment alone makes embeddings sensitive to task-irrelevant noise, while dropping the center loss yields diffuse regions. Jointly optimizing alignment and center losses isolates structural clusters without task bias, allowing subsequent cross-entropy to refine them. Conversely, simultaneous three-objective training forces label-driven rather than structure-driven clusters, while strictly sequential training suffers from representation drift.

5.5 STRUCTURE-AWARE EMBEDDING QUALITY (RQ4)

If teacher superiority depends on structural context, the embedding space (Sec. 4.1) should naturally group nodes favoring the same teacher into contiguous areas. Furthermore, our unsupervised region formation should discover these boundaries, and the assigned teachers should generalize to unseen nodes. Figure 2 validates this entirely on held-out test nodes. Figure 2 demonstrates that teacher superiority is structurally organized rather than scattered (**Left**: GAT dominates the largest area, followed by GRAPHSAGE, then GCN). Unsupervised nearest-center region assignment (**Middle**) closely aligns with these natural boundaries. Crucially, assigning teachers to regions based solely on training accuracy successfully recovers the true best teacher for unseen test nodes (**Right**), with minor disagreements confined to region boundaries. Because these regions are structural rather than class-driven, the embeddings serve as an effective, lightweight surrogate for expensive teacher forward passes during multi-teacher routing.

5.6 CALIBRATION (RQ5)

Knowledge distillation risks producing overconfident student models Yang & Song (2021), particularly conflict. We evaluate prediction reliability via ECE and NLL in Table 7. Notably, in heterophilic datasets MIDAS achieves better calibration than all individual teacher GNNs suggesting that structure-conditioned routing enhances prediction reliability as well. Removing the trunk-adapter causes both ECE and NLL to degrade sharply, confirming that the architecture is central to well-calibrated predictions.

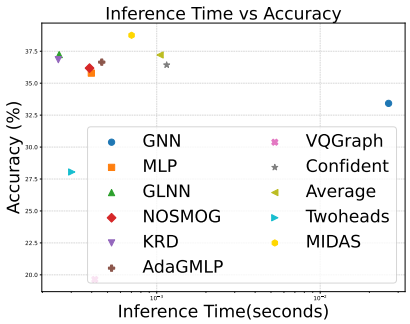
Table 6: Encoder ablation "+": trained jointly; "→": new step; Co-CS was observed to suffer from overfitting during the alignment training which led to sudden drop in performance in the first row

Schedule	Actor	PubMed	A-Ratings	Co-CS
Align only	36.45	79.32	46.85	66.41
Align → Align+Center	37.20	78.20	47.90	91.44
Align+Center+Ref throughout	37.50	80.90	48.65	92.37
Align → Center → Ref	36.57	79.45	48.20	91.67
Align → Align+Center → Align+Center+Ref	**38.75**	**81.70**	**49.46**	**93.10**

Table 7: Calibration (inductive): ECE (↓) and NLL (↓) vs. individual teachers. ‡:

Dataset	Metric	GRAPHSAGE	GAT	GCN	MIDAS
Actor	ECE	0.1577	0.0256	0.0263	**0.0234**
	NLL	1.5178	1.5502	1.5516	**1.3988**
	ECE/NLL‡		—		*0.712 / 1.582*
PubMed	ECE	0.0658	0.0452	0.0444	**0.0283**
	NLL	0.6334	0.5876	0.5709	**0.5329**
	ECE/NLL‡		—		*0.1398 / 0.7785*
Co-CS	ECE	0.0412	0.0389	0.0401	**0.0274**
	NLL	0.3821	0.3654	0.3712	**0.3398**
	ECE/NLL‡		—		*0.1563 / 1.2785*

Figure 3: Accuracy vs. inference time on PubMed (median of 20 passes, full test set).



5.7 LATENCY AND EFFICIENCY (RQ6)

We examine the trade-off between prediction accuracy and inference time on Pubmed, as shown in Figure 3. Although MIDAS incurs slightly higher latency than other MLP, it offers significantly better accuracy. Despite its multi-component design, MIDAS's inference time remains comparable to other MLP and is faster than multi-teacher ensembles, as it eliminates the need for the concurrent prediction from three MLP models. Given that predictive power is often the most critical requirement in production environments, this tradeoff is acceptable. The inference times in other datasets are comparable for all baselines except GRAPHSAGE, since they all utilize an MLP of the same parameter size. Overall, no other approach achieves the same balance of efficacy and latency as MIDAS. A detailed analysis of the space and time complexity is provided in App A.3. The core motivation of distillation is to enable deployment on resource constrained devices or mobile platforms where graph based message passing is computationally prohibitive. We report complexity analysis, inference statistics and training costs in A.3,A.7,A.14.

6 CONCLUSIONS

We introduced MIDAS, a multi-teacher GNN-to-MLP distillation framework built on one premise: no single GNN is the best teacher everywhere, and their relative strengths depends on structural context. MIDAS learns a feature-only embedding in which structural regions are discovered for unseen nodes, and supervises each region with its most accurate teacher through a dedicated adapter. GNN performance varies non-uniformly across structural regions.MIDAS achieves consistent improvements over all baselines across seven datasets with up to 20× inference speedup. Extending MIDAS to dynamic graphs is a direction for future work.

7 AI DISCLAIMERS

: We have used AI tools to aid and polish our writing, but all the ideas, experiments, and retrieval of prior work were done by the authors themselves.

REFERENCES

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *International conference on machine learning*, pp. 874–883. PMLR, 2020.

Yuan-Hung Chao, Chia-Hsun Lu, and Chih-Ya Shen. Transferring social network knowledge from multiple gnn teachers to kolmogorov-arnold networks. *arXiv preprint arXiv:2508.06663*, 2025.

Dexiong Chen, Leslie O'Bray, and Karsten Borgwardt. Structure-aware transformer for graph representation learning. In *International conference on machine learning*, pp. 3469–3489. PMLR, 2022.

Dawei Cheng, Yao Zou, Sheng Xiang, and Changjun Jiang. Graph neural networks for financial fraud detection: a review. *Frontiers of Computer Science*, 19(9):199609, 2025.

Mucong Ding, Kezhi Kong, Jingling Li, Chen Zhu, John Dickerson, Furong Huang, and Tom Goldstein. Vq-gnn: A universal framework to scale up graph neural networks using vector quantization. *Advances in Neural Information Processing Systems*, 34:6733–6746, 2021.

Shangchen Du, Shan You, Xiaojie Li, Jianlong Wu, Fei Wang, Chen Qian, and Changshui Zhang. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. *advances in neural information processing systems*, 33:12345–12355, 2020.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pp. 226–231. AAAI Press, 1996.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*, pp. 4116–4126. PMLR, 2020.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pp. 639–648, 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Chuanze Kang, Han Zhang, Zhuo Liu, Shenwei Huang, and Yanbin Yin. Lr-gnn: A graph neural network based on link representation for predicting molecular associations. *Briefings in Bioinformatics*, 23(1):bbab513, 2022.

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

Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. *Advances in Neural Information Processing Systems*, 33:4465–4478, 2020.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

Tianyu Liu, Yuge Wang, Rex Ying, and Hongyu Zhao. Muse-gnn: Learning unified gene representation from multimodal biological graph data. *Advances in neural information processing systems*, 36:24661–24677, 2023.

Weigang Lu, Ziyu Guan, Wei Zhao, and Yaming Yang. Adagmlp: Adaboosting gnn-to-mlp knowledge distillation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2060–2071, 2024.

Sudipta Modak, Aakarsh Malhotra, Sarthak Malik, Anil Surisetty, and Esam Abdel-Raheem. Cpawac: Constellation partitioning-based scalable weighted aggregation composition for knowledge graph embedding. In *International Joint Conference on Artificial Intelligence*, pp. 3504–3512, 2024.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4602–4609, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Harsh Pal, Sarthak Malik, Rajat Patel, and Aakarsh Malhotra. Structure aware latent embeddings for gnn to graph-free mlp distillation. In *International Joint Conference on Artificial Intelligence*, 2025.

Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287*, 2020.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations*.

Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 385–394, 2017.

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

Kartik Sharma, Yeon-Chang Lee, Sivagami Nambi, Aditya Salian, Shlok Shah, Sang-Wook Kim, and Srijan Kumar. A survey of graph neural networks for social recommender systems. *ACM Computing Surveys*, 56(10):1–34, 2024.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*, 2018.

Guangyuan Shi, Qimai Li, Wenlong Zhang, Jiaxin Chen, and Xiao-Ming Wu. Recon: Reducing conflicting gradients from the root for multi-task learning. *arXiv preprint arXiv:2302.11289*, 2023.

Yong-Min Shin and Won-Yong Shin. Propagate & distill: Towards effective graph learners using propagation-embracing mlps. *arXiv preprint arXiv:2311.17781*, 2023.

Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by subgraph reasoning. In *International conference on machine learning*, pp. 9448–9457. PMLR, 2020.

Yijun Tian, Chuxu Zhang, Zhichun Guo, Xiangliang Zhang, and Nitesh V Chawla. Nosmog: Learning noise-robust and structure-aware mlps on graphs. *arXiv preprint arXiv:2208.10010*, 2022.

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

Kai Wang, Yu Liu, Qian Ma, and Quan Z Sheng. Mulde: Multi-teacher knowledge distillation for low-dimensional knowledge graph embeddings. In *Proceedings of the Web Conference 2021*, pp. 1716–1726, 2021.

Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pp. 499–515. Springer, 2016.

Lirong Wu, Haitao Lin, Yufei Huang, and Stan Z Li. Quantifying the knowledge in gnns for reliable distillation into mlps. In *International Conference on Machine Learning*, pp. 37571–37581. PMLR, 2023.

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

Bo-Wei Yang, Ming-Yi Chang, Chia-Hsun Lu, and Chih-Ya Shen. Two heads are better than one: Teaching mlps with multiple graph neural networks via knowledge distillation. In *International Conference on Database Systems for Advanced Applications*, pp. 452–462. Springer, 2024.

Chuanguang Yang, Xinqiang Yu, Han Yang, Zhulin An, Chengqing Yu, Libo Huang, and Yongjun Xu. Multi-teacher knowledge distillation with reinforcement learning for visual recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9148–9156, 2025.

Lehan Yang and Jincen Song. Rethinking the knowledge distillation from the perspective of model calibration. *arXiv preprint arXiv:2111.01684*, 2021.

Ling Yang, Ye Tian, Minkai Xu, Zhongyi Liu, Shenda Hong, Wei Qu, Wentao Zhang, Bin Cui, Muhan Zhang, and Jure Leskovec. Vqgraph: Graph vector-quantization for bridging gnns and mlps. *arXiv preprint arXiv:2308.02117*, 2023.

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.

Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10737–10745, 2021.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. *Advances in neural information processing systems*, 33: 5812–5823, 2020.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

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

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. In *International Conference on Learning Representations*.

Chunhai Zhang, Jie Liu, Kai Dang, and Wenzheng Zhang. Multi-scale distillation from multiple graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 4337–4344, 2022a.

Hailin Zhang, Defang Chen, and Can Wang. Confidence-aware multi-teacher knowledge distillation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4498–4502. IEEE, 2022b.

Shichang Zhang, Yozen Liu, Yizhou Sun, and Neil Shah. Graph-less neural networks: Teaching old mlps new tricks via distillation. *arXiv preprint arXiv:2110.08727*, 2021a.

Wentao Zhang, Yuezihan Jiang, Yang Li, Zeang Sheng, Yu Shen, Xupeng Miao, Liang Wang, Zhi Yang, and Bin Cui. Rod: reception-aware online distillation for sparse graphs. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 2232–2242, 2021b.

Hongkuan Zhou, Ajitesh Srivastava, Hanqing Zeng, Rajgopal Kannan, and Viktor Prasanna. Accelerating large scale real-time gnn inference using channel pruning. *Proceedings of the VLDB Endowment*, 14(9):1597–1605, 2021.

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, 2020a.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020b.

A Appendix

A.1 Dataset Statistics

Table 8: Undirected graphs; h is edge homophily Zhu et al. (2020a).

Dataset	Nodes	Edges	Feat.	Cls.	h
Actor Pei et al. (2020)	7,600	30,019	932	5	0.22
PubMed Sen et al. (2008)	19,717	44,338	500	3	0.80
A-Photo Shchur et al. (2018)	7,650	119,081	745	8	0.83
Co-CS Shchur et al. (2018)	18,333	81,894	6,805	15	0.81
Flickr Zeng et al.	89,250	899,756	500	7	0.32
Questions Platonov et al.	48,921	153,540	301	2	0.84
A-Ratings Platonov et al.	24,492	93,050	300	5	0.38

A.2 Training Procedure

Algorithm 1 MIDAS: Structure-Aware Multi-Teacher Distillation

Require: Graph G, node features $\{\mathbf{x}_v\}$, teachers $\{f_t^k\}$
for each $v \in \mathcal{V}_{\text{train}}$ **do**
  $\mathbf{s}_v \leftarrow \text{Struc2Vec}(v)$
end for
for $e = 1$ **to** E_1 **do**
  Update: $\mathcal{L}_{\text{align}} + \lambda_1 \mathcal{L}_{\text{center}}$
end for
for $e = 1$ **to** E_2 **do**
  Update: $\mathcal{L}_{\text{align}} + \lambda_1 \mathcal{L}_{\text{center}} + \lambda_2 \mathcal{L}_{\text{CE}}$
end for
$\mathcal{C} \leftarrow \text{DBSCAN}(\{\mathbf{z}_v^{\text{feat}}\}, \epsilon, m)$
for each cluster $c_m \in \mathcal{C}$ **do**
  $t_m^* \leftarrow \text{argmax}_k \, \text{Acc}(f_t^k, c_m)$
end for
for $e = 1$ **to** E_3 **do**
  Update trunk: $\alpha_{\text{ce}} \mathcal{L}_{\text{CE}} + \alpha_{\text{fwd}} \mathcal{L}_{\text{KL}}$
end for
Freeze trunk ψ_θ
for $e = 1$ **to** E_4 **do**
  Update adapters: $\beta_{\text{ce}} \mathcal{L}_{\text{CE}} + \beta_{\text{dist}} \mathcal{L}_{\text{KL}} + \beta_{\text{cluster}} \mathcal{L}_{\text{cluster}}$
end for
for each test node v **do**
  $k^* \leftarrow \text{argmin}_m \, \|\mathbf{z}_v^{\text{feat}} - \mathbf{c}_m\|_2$
  return $\text{argmax}_c \; a_{\psi_{k^*}}(\psi_\theta(\mathbf{x}_v))$
end for

A.3 Complexity Analysis

Standard Struc2Vec structure descriptor is $\mathcal{O}(n^2 k')$ worst case The κ-NN approximation reduces this to $\mathcal{O}(n\kappa k')$ with speedup $\Theta(n/\kappa)$, ranging from $760\times$ to $8{,}952\times$ across our datasets. The dominant offline cost is Struc2Vec preprocessing; for large graphs, well-known κ-NN approximations reduce this to near-linear complexity with negligible accuracy. loss Ribeiro et al. (2017). The random walk phase (generating node sequences from M) contributes $\mathcal{O}(nRL)$ independently of the κ-NN approximation.This phase is parallel as each node's walks are independent, enabling

straightforward multi-core or GPU batching beyond the n/κ asymptotic gain. Dual-encoder training: $\mathcal{O}(E_{\text{dual}} \cdot |\mathcal{V}| \cdot d \cdot d')$. Student training: $\mathcal{O}(|\mathcal{V}| \cdot (E_1 H^2 + E_2 K H A))$.

Full pipeline complexity. Combining all phases:

$$T_{\text{MIDAS}} = \underbrace{\mathcal{O}(n\kappa k' \bar{d}^2)}_{\text{Struc2Vec (one-time)}} + \underbrace{\mathcal{O}(nRL)}_{\text{Walk generation}} + \underbrace{\mathcal{O}(E_{\text{dual}} \cdot n \cdot d \cdot d')}_{\text{Dual-Encoder}} + \underbrace{\mathcal{O}(n(E_1 H^2 + E_2 K H A))}_{\text{Student training}} \tag{7}$$

Inference per node is $\mathcal{O}(H^2 + HA)$, independent of graph size, since no adjacency matrix is accessed at test time. This is strictly faster than any GNN teacher requiring $\mathcal{O}(|\mathcal{E}| \cdot H^2)$ per forward pass.

A.4 Topological Alignment of Predictions (RQ5)

Table 9: Min-cut score of hard predictions (higher is better). GLNN is distilled from GraphSAGE; Gain is MIDAS over the best baseline.

Dataset	GraphSAGE	MLP	GLNN	MIDAS	Gain
A-Photo	0.9453	0.8738	0.9342	**0.9670**	+2.3%
PubMed	0.9607	0.8832	0.9432	**0.9654**	+0.5%
Co-CS	0.8582	0.7661	0.8789	**0.8912**	+1.4%

A fundamental problem in GNN-to-MLP distillation is the reconstruction of topology-related information solely from node features. To quantify this capability, we evaluate the model's ability to approximate the *min-cut problem* which partitions a set of N nodes in $\mathcal{V}$ into K disjoint subsets while minimizing the number of edges removed. Let $\boldsymbol{C}_k$ is the assignment matrix ($\boldsymbol{C}_k[i] = 1$ if v_i is in partition k), $\mathbf{A}$ is the adjacency matrix, and $\mathbf{D}$ is the degree matrix. Following Bianchi et al. (2020),Following Bianchi et al. (2020), we replace the discrete assignment $\boldsymbol{C}$ with the model's predicted distribution $\hat{\mathbf{Y}}$ to measure topological alignment.

$$\max_{\mathbf{S}} \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} \frac{\mathbf{S}_y^\top \mathbf{A} \mathbf{S}_y}{\mathbf{S}_y^\top \mathbf{D} \mathbf{S}_y}, \tag{8}$$

$$\text{MinCut}(\mathbf{S}) = \frac{\text{Tr}(\mathbf{S}^\top \mathbf{A} \mathbf{S})}{\text{Tr}(\mathbf{S}^\top \mathbf{D} \mathbf{S})} \in [0, 1]. \tag{9}$$

We evaluate this metric on the three homophilous datasets because a cut-based metric effectively measures Laplacian smoothness, a property that is only relevant when graph topology aligns with the label distributions, as is the case in homophilic settings. Table 9 shows that MIDAS achieves the highest min-cut values across the all three datasets, demonstrating that structure-aware distillation effectively encodes topological information through dual-encoder alignment and teacher-specific supervision.. This improvement is particularly pronounced on A-Photo and CS-CoAuthor datasets, where the underlying topology is more complex. In contrast, pure MLP performs poorly, confirming the difficulty of inferring topology from node features alone. GLNN improves over MLP through learned graph approximations, while SAGE benefits from explicit neighborhood aggregation.

A.5 Transductive Results

Table 10 reports the full transductive results for individual-teacher, multi-teacher, and feature-only baselines. Although the evaluation is transductive, MIDAS's student uses only node features for placement and prediction at inference. MIDAS achieves the highest reported score on Actor, PubMed, Flickr, and Questions; the strongest results on the remaining datasets vary by method.

Table 10: Transductive setting: accuracy (%), $\mu \pm \sigma$ over 10 seeds; Questions* reports ROC-AUC (%).

Teacher	Method	Actor	PubMed	A-Photo	Co-CS	Flickr	Questions*	A-Ratings
GCN	Standard	29.09±0.39	79.88±0.52	91.84±0.54	91.73±0.37	49.97±0.39	65.81±2.66	48.78±0.41
	GLNN	31.31±0.44	76.66±0.30	90.31±0.45	86.48±0.33	48.94±0.26	68.18±1.37	48.91±0.25
	NOSMOG	30.53±0.40	77.11±0.34	90.69±0.24	87.68±0.44	49.17±0.15	69.58±0.17	49.44±0.23
	KRD	30.05±0.34	77.90±0.35	91.03±0.35	88.70±0.33	49.33±0.41	68.97±1.23	49.12±0.45
	AdaGMLP	29.92±0.44	78.62±0.29	91.31±0.39	88.93±0.36	49.91±0.23	72.56±0.49	49.36±0.32
	VQGraph	29.79±0.34	78.18±0.49	91.39±0.29	89.80±0.39	49.55±0.48	73.01±0.62	49.28±0.45
	SALE-MLP	35.83±1.01	79.47±2.35	90.52±1.76	89.90±3.29	48.69±0.26	71.01±0.62	48.15±0.13
GAT	Standard	30.20±0.57	80.09±0.25	92.84±0.17	92.50±0.39	51.02±0.24	70.87±1.42	49.14±0.42
	GLNN	32.42±0.55	77.01±0.30	90.58±0.51	87.16±0.45	49.08±0.23	70.66±0.82	49.41±0.34
	NOSMOG	31.89±0.16	78.22±0.19	91.26±0.50	87.56±0.29	49.81±0.26	73.96±0.91	49.60±0.44
	KRD	31.14±0.55	78.31±0.27	91.71±0.17	88.78±0.35	49.47±0.61	74.56±1.23	49.57±0.25
	AdaGMLP	31.44±0.50	78.33±0.51	91.92±0.51	89.82±0.42	49.59±0.33	72.18±0.59	49.50±0.29
	VQGraph	31.15±0.22	79.31±0.45	92.78±0.25	89.97±0.26	50.07±0.21	75.01±2.07	49.39±0.48
	SALE-MLP	35.03±0.64	79.42±1.13	90.57±1.85	87.38±2.13	48.77±0.33	74.43±1.17	49.05±0.25
GRAPHSAGE	Standard	34.26±0.32	80.71±0.25	**93.69**±0.41	**94.27**±0.53	51.49±0.27	75.48±0.83	49.31±0.46
	GLNN	34.98±0.32	77.91±0.65	90.96±0.43	86.79±0.38	50.08±0.21	75.18±0.43	49.99±0.39
	NOSMOG	34.87±0.40	78.37±0.42	92.05±0.19	88.55±0.40	50.32±0.27	76.01±1.12	50.09±0.19
	KRD	34.59±0.41	78.81±0.45	92.29±0.47	89.60±0.64	50.56±0.12	76.34±0.87	49.72±0.42
	AdaGMLP	34.75±0.18	79.29±0.23	92.16±0.37	89.72±0.33	50.10±0.30	77.01±1.60	49.56±0.35
	VQGraph	34.41±0.25	79.86±0.36	92.57±0.39	90.80±0.32	50.82±0.62	77.53±0.93	49.46±0.38
	SALE-MLP	35.69±1.49	79.85±2.13	90.33±0.53	89.91±3.00	48.68±0.26	76.12±1.15	48.89±0.33
–	MLP	36.91±0.52	69.36±0.44	85.20±0.38	70.23±0.61	46.20±0.35	70.40±0.74	47.96±0.33
All three	Mean	35.20±0.39	80.10±0.45	92.80±0.35	92.40±0.26	51.80±0.31	75.12±2.87	48.50±0.49
	Confidence	36.10±0.48	80.11±0.48	92.95±0.38	92.70±0.35	52.00±0.23	77.32±1.56	48.70±0.50
	Two-heads	36.80±0.44	80.60±0.27	92.10±0.27	93.00±0.37	52.20±0.22	76.72±1.60	48.90±0.44
	Soft MoE	37.20±0.49	81.10±0.36	92.30±0.24	93.20±0.40	52.40±0.48	73.29±1.81	49.00±0.36
	AE-KD	35.66±1.57	78.56±1.35	90.43±1.88	93.19±0.61	48.19±0.37	77.81±1.50	48.11±0.27
	CA-MKD	33.92±0.71	78.94±1.50	90.86±2.02	92.95±0.71	51.17±1.37	76.65±2.07	47.51±0.67
	MTKD-RL	35.23±0.32	78.18±1.06	92.24±1.22	93.01±0.72	48.75±1.40	75.91±1.11	48.12±0.56
MIDAS (ours)		**39.10**†±0.41	**81.90**†±0.38	93.57±0.29	93.52±0.31	**52.75**±0.44	**78.57**±1.81	49.30±0.35

A.6 MULTI TEACHER ADVANTAGE: THEORETICAL ANALYSIS

Proposition 1 (Optimality of Cluster-Routed Distillation)**.** *Let G have L structural clusters $\{c_1, \ldots, c_L\}$ with cluster sizes $|c_l|$ and total $n = \sum_l |c_l|$. For teacher pool $\mathcal{T} = \{T_1, \ldots, T_K\}$, define the cluster-optimal teacher $T_l^* = \arg\min_{T_k \in \mathcal{T}} R(T_k, c_l)$, where $R(T_k, c_l)$ is the distillation risk of teacher T_k on cluster c_l. Then for any fixed single teacher $T_j \in \mathcal{T}$:*

$$\underbrace{\sum_{l=1}^{L} \frac{|c_l|}{n} R(T_j, c_l)}_{R_{\text{single}}(T_j)} \;\geq\; \underbrace{\sum_{l=1}^{L} \frac{|c_l|}{n} R(T_l^*, c_l)}_{R^*_{\text{multi}}} \tag{10}$$

with strict inequality whenever $T_l^* \neq T_{l'}^*$ for some $l \neq l'$.

Proof. By definition of T_l^*, we have $R(T_l^*, c_l) \leq R(T_j, c_l)$ for every l and every $T_j \in \mathcal{T}$. Multiplying by the non-negative weight $|c_l|/n$ and summing gives Eq. (10). If $T_l^* \neq T_{l'}^*$ for some pair, then no single T_j can equal T_l^* for all l simultaneously, and at least one term is strict. □

.

A.7 INFERENCE LATENCY IN CPU IN MILLISECONDS

MIDAS achieves 7–22× speedup over teacher GNNs and a 13–22× speedup over running all three teachers sequentially in a CPU.

longtable

Table 11: Inference latency (CPU, milliseconds).

Method	Actor	PubMed	Co-CS	A-Photo
Avg. teacher	47.0	88.7	371.5	105.5
Sequential ensemble	141.7	266.1	1114.5	316.5
MIDAS student	6.61	20.3	51.1	15.7
Speedup vs avg. teacher	7.1×	4.4×	7.3×	6.7×
Speedup vs ensemble	21.4×	13.1×	21.8×	20.2×

Table 12: Notation used across the appendix propositions and theorems.

Symbol	Meaning
$T^k(v)$	Teacher k's output distribution at node v
Δ^{C-1}	The probability simplex over C classes
$\bar{T}(v)$	Pointwise average of the K teachers, $\frac{1}{K}\sum_k T^k(v)$
$s(v)$	A candidate student distribution at v
$J(v)$	Teacher diversity at v: $\frac{1}{K}\sum_k \mathrm{KL}(T^k(v) \,\|\, \bar{T}(v))$, independent of s
$T_k^{\mathrm{rest}}(v)$	Average of the $K-1$ teachers excluding k *i.e.* $\frac{1}{K-1}\sum_{j\neq k} T^j(v)$
$q_{\hat{\theta}}(v)$	Output distribution of the frozen, Phase1 trained trunk at v
c_l	Node set of region l ($=\{v : \pi(v) = l\}$)
$R_1^{(l)}(\hat{\theta})$	Phase1 risk in region l *i.e.* $\mathbb{E}_{v\sim c_l}\left[\mathrm{KL}(\bar{T}(v) \,\|\, q_{\hat{\theta}}(v))\right]$
$\bar{d}_l$	Region averaged teacher disagreement: $\mathbb{E}_{v\sim c_l}[d_{\mathrm{TV}}(T^*, T_*^{\mathrm{rest}})]$
z_i^f	Shorthand for $z_{v_i}^{\mathrm{feat}}$, node i's feature embedding, $\|z_i^f\|_2 = 1$
z_i^s	Shorthand for $z_{v_i}^{\mathrm{struct}}$, node i's structural embedding, $\|z_i^s\|_2 = 1$
$\bar{z}^s$	Batch mean of structural embeddings: $\frac{1}{N}\sum_j z_j^s$
$\tilde{z}_i^s$	Batch mean centered structural residual: $z_i^s - \bar{z}^s$
d_i	Node i's cross-view gap: $\|z_i^f - \tilde{z}_i^s\|_2$
$\bar{M}$	Average squared cross-view gap: $\frac{1}{N}\sum_i d_i^2$
n_l	Number of training nodes in region l
$w_{\min}$	Smallest region's training share: $\min_l n_l/N$
$\pi(i)$	Shorthand for $\pi(v_i)$, node i's region label
$\mathbf{C}_l^s$	Region l's structural centroid: $\frac{1}{n_l}\sum_{\pi(i)=l} \tilde{z}_i^s$
$\mathbf{C}_l^f$	Region l's feature centroid: $\frac{1}{n_l}\sum_{\pi(i)=l} z_i^f$
$\pi^s(i)$	Graph based router: nearest structural centroid to node i
$\pi^f(i)$	Graph free router: nearest feature centroid to node i
γ_s	Structural routing margin (smallest lead over the closest competitor, over all nodes)

A.8 Lower Bound on Alignment Loss and a Cluster Misrouting Bound

Proposition 2 (Alignment Loss Lower Bound via the Centered Structural Residual)**.** *Let* $z_i^f := z_{v_i}^{\mathrm{feat}}$, $z_i^s := z_{v_i}^{\mathrm{struct}}$, *with* $\|z_i^f\|_2 = \|z_i^s\|_2 = 1$ *for all* i, *let* $\bar{z}^s := \frac{1}{N}\sum_{j=1}^N z_j^s$, *let* $\tilde{z}_i^s := z_i^s - \bar{z}^s$ *denote the batch-mean-centered structural residual, Then*

$$\mathcal{L}_{\mathrm{align}} \;\geq\; \log N \;-\; \frac{1}{2\tau} \;-\; \frac{1}{2N\tau}\sum_{i=1}^N \|\tilde{z}_i^s\|_2^2 \;+\; \frac{1}{2N\tau}\sum_{i=1}^N \|z_i^f - \tilde{z}_i^s\|_2^2.$$

Proof.

$$
\begin{aligned}
\mathcal{L}_{\text{align}} &= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp\big((z_i^f)^\top z_i^s/\tau\big)}{\sum_{j=1}^{N}\exp\big((z_i^f)^\top z_j^s/\tau\big)} \\
&= -\frac{1}{N}\sum_{i=1}^{N}\frac{(z_i^f)^\top z_i^s}{\tau} \;+\; \frac{1}{N}\sum_{i=1}^{N}\log\sum_{j=1}^{N}\exp\left(\frac{(z_i^f)^\top z_j^s}{\tau}\right) \\
&\geq -\frac{1}{N}\sum_{i=1}^{N}\frac{(z_i^f)^\top z_i^s}{\tau} \;+\; \log N \;+\; \frac{1}{N}\sum_{i=1}^{N}\frac{1}{N}\sum_{j=1}^{N}\log\exp\left(\frac{(z_i^f)^\top z_j^s}{\tau}\right) \\
&= -\frac{1}{N}\sum_{i=1}^{N}\frac{(z_i^f)^\top z_i^s}{\tau} \;+\; \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{(z_i^f)^\top z_j^s}{\tau} \;+\; \log N \\
&= -\frac{1}{N}\sum_{i=1}^{N}\frac{(z_i^f)^\top z_i^s}{\tau} \;+\; \frac{1}{N}\sum_{i=1}^{N}\frac{(z_i^f)^\top}{\tau}\left(\frac{1}{N}\sum_{j=1}^{N}z_j^s\right) \;+\; \log N \\
&= -\frac{1}{N}\sum_{i=1}^{N}\frac{(z_i^f)^\top z_i^s}{\tau} \;+\; \frac{1}{N}\sum_{i=1}^{N}\frac{(z_i^f)^\top \bar{z}^s}{\tau} \;+\; \log N; \quad \bar{z}^s := \frac{1}{N}\sum_{j=1}^{N}z_j^s \\
&= \log N \;-\; \frac{1}{N}\sum_{i=1}^{N}\frac{(z_i^f)^\top(z_i^s-\bar{z}^s)}{\tau} \\
&= \log N \;-\; \frac{1}{N\tau}\sum_{i=1}^{N}(z_i^f)^\top\tilde{z}_i^s; \qquad \tilde{z}_i^s := z_i^s-\bar{z}^s \\
&= \log N - \frac{1}{2\tau} - \frac{1}{2N\tau}\sum_{i=1}^{N}\|\tilde{z}_i^s\|^2 + \frac{1}{2N\tau}\sum_{i=1}^{N}\|z_i^f-\tilde{z}_i^s\|^2
\end{aligned}
$$

□

Corollary 1 (Shrinkage of MSE Ceiling). *Let* $\bar{S} := \frac{1}{N}\sum_{i=1}^{N}\|\tilde{z}_i^s\|_2^2$ *and* $\bar{M} := \frac{1}{N}\sum_{i=1}^{N}\|z_i^f-\tilde{z}_i^s\|_2^2$. *Then*

$$\bar{M} \;\leq\; 2\tau\big(\mathcal{L}_{\text{align}} - \log N\big) \;+\; 1 \;+\; \bar{S}.$$

Notice that, as $\mathcal{L}_{\text{align}}$ decreases during training, $\bar{M}$ shrinks

Theorem 1 (Cluster-Misrouting Bound from the Alignment Loss). *Fix the (non-noise) training partition* π *with regions* $l = 1,\dots,L$ *of sizes* n_l, *and let* $w_{\min} := \min_l n_l/N$. *Following the shorthand notations* $z_i^f := z_{v_i}^{\text{feat}}$, $z_i^s := z_{v_i}^{\text{struct}}$, $\pi(i) := \pi(v_i)$, *and define the paired region centroids and the per-node cross-view discrepancy* $\mathbf{C}_l^s := \frac{1}{n_l}\sum_{\pi(i)=l}\tilde{z}_i^s$, $\mathbf{C}_l^f := \frac{1}{n_l}\sum_{\pi(i)=l}z_i^f$, $d_i := \big\|z_i^f-\tilde{z}_i^s\big\|_2$, $\bar{M} := \frac{1}{N}\sum_{i=1}^{N}d_i^2$ *where* $\tilde{z}_i^s := z_i^s-\bar{z}^s$ *is the batch centered structural residual of Proposition 2, so that* $\bar{M}$ *is exactly the quantity bounded by Corollary 1. Define the nearest centroid routing rules as* $\pi^s(i) := \operatorname{argmin}_l\big\|\tilde{z}_i^s-\mathbf{C}_l^s\big\|_2$, $\pi^f(i) := \operatorname{argmin}_l\big\|z_i^f-\mathbf{C}_l^f\big\|_2$, *and the structural routing margin* $\gamma_s := \min_i\ \min_{l'\neq\pi^s(i)}\Big(\big\|\tilde{z}_i^s-\mathbf{C}_{l'}^s\big\|_2-\big\|\tilde{z}_i^s-\mathbf{C}_{\pi^s(i)}^s\big\|_2\Big) > 0$ *Let* $\varepsilon := \sqrt{\bar{M}/w_{\min}}$, *and assume* $\gamma_s > 2\varepsilon$. *Then, for a node* v *drawn uniformly from the training nodes, we get*

$$\Pr\big[\pi^f(v)\neq\pi^s(v)\big] \;\leq\; \frac{4\,\bar{M}}{(\gamma_s-2\varepsilon)^2}.$$

Moreover, substituting the ceiling of Corollary 1, $\bar{M} \le \left[2\tau(\mathcal{L}_{\text{align}} - \log N) + 1 + \bar{S}\right]_+ =: \bar{M}_\circ$ *(valid provided* $\gamma_s > 2\sqrt{\bar{M}_\circ / w_{\min}}$*, a strictly stronger requirement than* $\gamma_s > 2\varepsilon$*),*

$$\Pr\left[\pi^f(v) \neq \pi^s(v)\right] \;\le\; \frac{4\,\bar{M}_\circ}{\left(\gamma_s - 2\sqrt{\bar{M}_\circ / w_{\min}}\right)^2}.$$

Proof. For every region l, by the triangle inequality and Cauchy–Schwarz,

$$\left\|\mathbf{C}_l^f - \mathbf{C}_l^s\right\|_2 = \left\|\frac{1}{n_l}\sum_{\pi(i)=l}\left(z_i^f - \tilde{z}_i^s\right)\right\|_2 \le \frac{1}{n_l}\sum_{\pi(i)=l} d_i \le \sqrt{\frac{1}{n_l}\sum_{\pi(i)=l} d_i^2} \le \sqrt{\frac{1}{n_l}\sum_{i=1}^{N} d_i^2} = \sqrt{\frac{N\bar{M}}{n_l}} \;\le\; \sqrt{\frac{\bar{M}}{w_{\min}}}$$

This holds for every l, so $\sup_l \|\mathbf{C}_l^f - \mathbf{C}_l^s\|_2 \le \varepsilon$.

Now, Fix node i. For the true region, two applications of the triangle inequality together with Step 1 give

$$\left\|z_i^f - \mathbf{C}^f_{\pi^s(i)}\right\|_2 \;\le\; \left\|z_i^f - \tilde{z}_i^s\right\|_2 + \left\|\tilde{z}_i^s - \mathbf{C}^s_{\pi^s(i)}\right\|_2 + \left\|\mathbf{C}^s_{\pi^s(i)} - \mathbf{C}^f_{\pi^s(i)}\right\|_2 \;\le\; d_i + \left\|\tilde{z}_i^s - \mathbf{C}^s_{\pi^s(i)}\right\|_2 + \varepsilon,$$

while for any competitor region $l' \neq \pi^s(i)$, similar argument shows that

$$\left\|\tilde{z}_i^s - \mathbf{C}^s_{l'}\right\|_2 \le \left\|\tilde{z}_i^s - z_i^f\right\| + \left\|z_i^f - \mathbf{C}^f_{l'}\right\| + \left\|\mathbf{C}^f_{l'} - \mathbf{C}^s_{l'}\right\| \le d_i + \left\|z_i^f - \mathbf{C}^f_{l'}\right\| + \varepsilon$$

Subtracting, above two inequalities, we get,

$$\begin{aligned}\left\|z_i^f - \mathbf{C}^f_{l'}\right\|_2 - \left\|z_i^f - \mathbf{C}^f_{\pi^s(i)}\right\|_2 &\ge \left\|\tilde{z}_i^s - \mathbf{C}^s_{l'}\right\|_2 - \left\|\tilde{z}_i^s - \mathbf{C}^s_{\pi^s(i)}\right\|_2 \;-\; 2(d_i + \varepsilon) \quad \forall\, l' \neq \pi^s(i) \\ &\ge \gamma_s - 2(d_i + \varepsilon)\end{aligned}$$

Therefore, whenever $\gamma_s > 2(d_i + \varepsilon)$, i.e. whenever $d_i \;<\; \frac{\gamma_s}{2} - \varepsilon \;=:\; t$ Under this condition, $\pi^f(i) = \pi^s(i)$. Equivalently, by contraposition, $\pi^f(i) \neq \pi^s(i) \implies d_i \;\ge\; t$

Now, Since $\gamma_s > 2\varepsilon$ by hypothesis, $t := \gamma_s/2 - \varepsilon > 0$. Let v be drawn uniformly at random from the N training nodes, so that d_v^2 is a non-negative random variable we get the following using Markov inequality

$$\Pr\left[d_v \ge t\right] \;=\; \Pr\left[d_v^2 \ge t^2\right] \;\le\; \frac{\mathbb{E}[d_v^2]}{t^2} \;=\; \frac{\sum_{i=1}^N d_i^2}{Nt^2} \;=\; \frac{\bar{M}}{t^2}.$$

Combining with the contrapositive argument we get, ($\pi^f(v) \neq \pi^s(v) \Rightarrow d_v \ge t$),

$$\Pr\left[\pi^f(v) \neq \pi^s(v)\right] \;\le\; \Pr\left[d_v \ge t\right] \;\le\; \frac{\bar{M}}{t^2} \;=\; \frac{\bar{M}}{(\gamma_s/2 - \varepsilon)^2} \;=\; \frac{4\bar{M}}{(\gamma_s - 2\varepsilon)^2}.$$

Substituting $\bar{M} \le \bar{M}_\circ$ from Corollary 1, provided the stated positivity condition $\gamma_s > 2\sqrt{\bar{M}_\circ / w_{\min}}$ holds, gives

$$\Pr\left[\pi^f(v) \neq \pi^s(v)\right] \;\le\; \frac{4\,\bar{M}_\circ}{\left(\gamma_s - 2\sqrt{\bar{M}_\circ / w_{\min}}\right)^2}.$$

□

A.9 Decomposition of Phase-1 Loss into Teacher Diversity and Fit to the Average Teacher

Proposition 3 (Phase-1 Objective Learns the Average Teacher)**.** *Let* $T^1, \ldots, T^K : \mathcal{V} \to \Delta^{C-1}$ *be the* K *teacher output distributions, where* Δ^{C-1} *denotes the probability simplex, let* $\bar{T}(v) := \frac{1}{K}\sum_{k=1}^K T^k(v)$ *denote their pointwise average, and let* $s(v) \in \Delta^{C-1}$ *be any candidate student*

distribution. Then for every node v, where $J(v) := \frac{1}{K}\sum_{k=1}^{K} \mathrm{KL}\big(T^k(v) \,\|\, \bar{T}(v)\big)$ is the (Jensen Shannon) teacher diversity at v, independent of s.

$$\frac{1}{K}\sum_{k=1}^{K} \mathrm{KL}\big(T^k(v) \,\|\, s(v)\big) \;=\; J(v) \;+\; \mathrm{KL}\big(\bar{T}(v) \,\|\, s(v)\big), \tag{11}$$

Consequently, $\mathcal{L}_{\mathrm{phase1}}$ (Eq.6) is minimized over s exactly at $s(v) = \bar{T}(v)$ for every v, with irreducible loss floor $\mathbb{E}_v[J(v)]$.

Proof. Fix v and expand the left-hand side of (11) coordinate-wise:

$$\begin{aligned}
\frac{1}{K}\sum_{k=1}^{K} \mathrm{KL}(T^k \| s) &= \frac{1}{K}\sum_{k=1}^{K}\sum_{c=1}^{C} T^k(c) \log \frac{T^k(c)}{s(c)} \\
&= \frac{1}{K}\sum_{k=1}^{K}\sum_{c=1}^{C} T^k(c) \left[\log \frac{T^k(c)}{\bar{T}(c)} + \log \frac{\bar{T}(c)}{s(c)}\right] \\
&= \underbrace{\frac{1}{K}\sum_{k=1}^{K}\sum_{c=1}^{C} T^k(c) \log \frac{T^k(c)}{\bar{T}(c)}}_{=J(v)} \;+\; \underbrace{\sum_{c=1}^{C}\left(\frac{1}{K}\sum_{k=1}^{K} T^k(c)\right) \log \frac{\bar{T}(c)}{s(c)}}_{=\mathrm{KL}(\bar{T}(v)\|s(v))},
\end{aligned} \tag{12}$$

Notice that, when the right-hand side of (11) is minimized over s iff $\mathrm{KL}(\bar{T}(v)\|s(v)) = 0$, i.e. $s(v) = \bar{T}(v)$ (by non-negativity and definiteness of KL divergence), at which point the left-hand side attains its minimum value $J(v)$. Averaging (11) over $v \in \mathcal{V}_{\mathrm{train}}$ gives $\mathcal{L}_{\mathrm{phase1}}(s) = \mathbb{E}_v[J(v)] + \mathbb{E}_v[\mathrm{KL}(\bar{T}(v)\|s(v))] \geq \mathbb{E}_v[J(v)]$, with equality iff $s(v) = \bar{T}(v)$ □

Distance Propagation: Phase1 Training Loss Bounds Phase2's Starting Distance

Proposition 4 (Bound on $\mathrm{KL}(T^k(v)\|\bar{T}(v))$)**.** *Let $T^k(v)$ denote the kth teacher's output at node v, let $T_k^{\mathrm{rest}}(v) := \frac{1}{K-1}\sum_{j\neq k} T^j(v)$ denote the average of the remaining $K-1$ teachers $\bar{T}(v) = \frac{1}{K}\sum_{j=1}^{K} T_j(v)$ as before. Then $\mathrm{KL}\big(T^k \,\|\, \bar{T}\big) \;\leq\; \log K \cdot d_{\mathrm{TV}}\big(T^k,\, T_k^{\mathrm{rest}}\big)$.*

Proof. For each class c with $T_c^k > 0$, we define $\rho_c := T_{kc}^{\mathrm{rest}}/T_c^k$. Then

$$\bar{T}_c \;=\; \frac{1}{K}T_c^k + \frac{K-1}{K}T_{kc}^{\mathrm{rest}} \;=\; \frac{1}{K}T_c^k\big(1 + (K-1)\rho_c\big), \text{then,}$$

$$\mathrm{KL}(T^k\|\bar{T}) \;=\; \sum_c T_c^k \log \frac{T_c^k}{\bar{T}_c} \;=\; \sum_c T_c^k \log \frac{K}{1+(K-1)\rho_c} \;=\; \log K \;-\; \mathbb{E}_{c\sim T^k}\big[f(\rho_c)\big],$$

where $f(\rho) := \log\big(1 + (K-1)\rho\big)$. Notice, $f(\rho)$ is a concave and increasing function, therefore

$$f(\rho) = f(\rho\cdot 1 + (1-\rho)\cdot 0) \;\geq\; \rho\, f(1) + (1-\rho)\, f(0) \;=\; \rho \log K, \qquad \rho \in [0,1].$$

Moreover, Since f is increasing, $f(\rho) \geq f(1) = \log K$ for $\rho \geq 1$.

$$f(\rho) \;\geq\; \begin{cases} \rho \log K & \rho \in [0,1] \\ \log K & \rho > 1 \end{cases} = \log K \cdot \min(\rho, 1) \quad \forall \rho \geq 0.$$

Taking expectation over $c \sim T^k$,

$$\mathbb{E}_{c\sim T^k}\big[f(\rho_c)\big] \;\geq\; \log K \cdot \mathbb{E}_{c\sim T^k}\big[\min(\rho_c, 1)\big].$$

Since $T_c^k \cdot \min(\rho_c, 1) = \min(T_{kc}^{\mathrm{rest}},\, T_c^k)$,

$$\mathbb{E}_{c\sim T^k}\big[\min(\rho_c, 1)\big] \;=\; \sum_c \min\big(T_{kc}^{\mathrm{rest}},\, T_c^k\big) \;=\; 1 - d_{\mathrm{TV}}\big(T^k, T_k^{\mathrm{rest}}\big),$$

(using the standard identity $\sum_c \min(p_c, q_c) = 1 - d_{\mathrm{TV}}(p, q)$)

Combining the expression of expectation and $\mathrm{KL}(T^k\|\bar{T})$, we get the desired result

$$\mathrm{KL}(T^k\|\bar{T}) \;=\; \log K - \mathbb{E}_{c\sim T^k}[f(\rho_c)] \;\leq\; \log K \cdot d_{\mathrm{TV}}\big(T^k, T_k^{\mathrm{rest}}\big).$$

As a direct consequence of proposition 4, we can show that $\mathrm{KL}(T^*\|\bar{T}) \leq \log K \cdot d_{\mathrm{TV}}(T^*, T_*^{\mathrm{rest}})$ for the optimal teacher T^* □

Theorem 2 (Phase-1-to-Phase-2 Distance Propagation)**.** *Let $q_{\hat{\theta}}(v)$ denote the output distribution after Phase1 training at node v, and let $R_1^{(l)}(\hat{\theta}) := \mathbb{E}_{v\sim c_l}\big[\mathrm{KL}(\bar{T}(v)\,\|\,q_{\hat{\theta}}(v))\big]$ denote Phase1's population risk restricted to cluster l, and $\bar{d}_l := \mathbb{E}_{v\sim c_l}\big[d_{\mathrm{TV}}\big(T^*,\,T^{\mathrm{rest}}\big)\big]$ be the cluster averaged disagreement between the assigned teacher and the average of the other $K-1$ teachers. Then,*

$$\mathbb{E}_{v\sim c_l}\Big[\,d_{\mathrm{TV}}\big(T_l^*(v),\,q_{\hat{\theta}}(v)\big)\,\Big] \;\le\; \frac{1}{\sqrt{2}}\left[\sqrt{\log K\cdot\bar{d}_l}\;+\;\sqrt{R_1^{(l)}(\hat{\theta})}\right].$$

Proof. Fix $v\in c_l$. By the triangle inequality of d_{TV} with the average teacher $\bar{T}(v)$ as the intermediate point,

$$\begin{aligned}
d_{\mathrm{TV}}\big(T_l^*(v),\,q_{\hat{\theta}}(v)\big) &\le d_{\mathrm{TV}}\big(T_l^*(v),\,\bar{T}(v)\big)\;+\;d_{\mathrm{TV}}\big(\bar{T}(v),\,q_{\hat{\theta}}(v)\big).\\
&\le \sqrt{\frac{\mathrm{KL}\big(T_l^*(v)\,\|\,\bar{T}(v)\big)}{2}}\;+\;\sqrt{\frac{\mathrm{KL}\big(\bar{T}(v)\,\|\,q_{\hat{\theta}}(v)\big)}{2}}\quad\text{(Pinsker's Inequality)}\\
&\le \sqrt{\frac{\log K\cdot d_{\mathrm{TV}}\big(T^*,\,T^{\mathrm{rest}}\big)}{2}}\;+\;\sqrt{\frac{\mathrm{KL}\big(\bar{T}(v)\,\|\,q_{\hat{\theta}}(v)\big)}{2}}\quad\text{(proposition4)}\\
\mathbb{E}_{v\sim c_l}\big[d_{\mathrm{TV}}\big(T_l^*(v),\,q_{\hat{\theta}}(v)\big)\big] &\le \mathbb{E}_{v\sim c_l}\left[\sqrt{\frac{\log K\cdot d_{\mathrm{TV}}\big(T^*,\,T^{\mathrm{rest}}\big)}{2}}\right]\;+\;\mathbb{E}_{v\sim c_l}\left[\sqrt{\frac{\mathrm{KL}\big(\bar{T}(v)\,\|\,q_{\hat{\theta}}(v)\big)}{2}}\right]\\
&\le \sqrt{\frac{\log K\cdot\mathbb{E}_{v\sim c_l}\big[d_{\mathrm{TV}}\big(T^*,\,T^{\mathrm{rest}}\big)\big]}{2}}+\sqrt{\frac{\mathbb{E}_{v\sim c_l}\big[\mathrm{KL}(\bar{T}(v)\,\|\,q_{\hat{\theta}}(v))\big]}{2}}\\
&\le \sqrt{\frac{\log K\cdot\bar{d}_l}{2}}+\sqrt{\frac{R_1^{(l)}(\hat{\theta})}{2}}
\end{aligned}$$

□

A.10 Hardware and Software Environment

All accuracy numbers are averaged over ten independent seeds. Training times are measured on the A100; CPU latency figures are additionally reported to characterise inference behaviour independent of GPU availability (both are in Table 18).

Table 13: Hardware and software environment.

Framework	PyTorch 1.12.0, Python 3.11, DGL
Training device	NVIDIA A100-SXM4-80GB, CUDA 12.2
CPU device	Apple M4 Max

A.11 Student and Teacher Network Configurations

Table 14 lists per-dataset hidden dimensions, layer counts, and feature mask settings. S denotes the student MLP; T denotes the teacher GNN (shared architecture across GCN, GAT, and GraphSAGE). *Masks* is the number of feature masks applied during training where dataset splits are not pre-defined; "–" indicates a standard random split is used instead (80/10/10 train/validation/test).

A.12 DBSCAN Clusters Discovered per Dataset

Table 15 reports the final cluster count and the ϵ value selected via grid search on the validation set. Embeddings are ℓ_2-normalised prior to DBSCAN, so all pairwise distances lie in $[0, 2]$.

Table 14: Student and teacher network configurations.

Dataset	S.Dim	S.Layers	T.Dim	T.Layers	Masks
Actor	128	2	256	3	10
PubMed	256	3	256	3	–
Amazon-Photo	256	2	256	2	–
Coauthor CS	256	2	256	2	–
Flickr	512	3	2048	3	–
Questions	256	2	256	3	10
Amazon Ratings	256	2	256	3	10

Table 15: DBSCAN clusters discovered per dataset.

Dataset	# Clusters (L)	ϵ
Actor	3	0.30
PubMed	4	0.25
Amazon-Photo	5	0.30
Coauthor CS	6	0.25
Flickr	4	0.30
Questions	3	0.25
Amazon Ratings	4	0.30

A.13 DBSCAN ϵ Sensitivity Analysis

ϵ controls cluster granularity, not the number of adapters: the number of cluster-specific adapters equals the number of teachers (3), which is fixed. What ϵ governs is how finely the node population is partitioned before teacher assignment — a smaller ϵ yields more, tighter clusters, while a larger ϵ merges them. Sweeping ϵ on Actor and PubMed (Tables 16–17) reveals two regimes: below a dataset-specific threshold, most nodes are labelled outliers and accuracy drops noticeably; above it, accuracy variance stays below 0.008, indicating a stable operating regime. The ϵ selected for every dataset (Table 15) falls within this stable region. For a new dataset, we recommend sweeping $\epsilon \in [0.2, 0.7]$ in steps of 0.05, targeting an outlier ratio below 15% and at least two non-trivial clusters; the silhouette score is a reliable guide when ground-truth labels are unavailable.

Table 16: ϵ sweep on Actor.

ϵ	# Clusters	Outlier %	Test Acc.
0.10	8	41.2	33.81
0.15	6	28.7	35.40
0.20	5	17.3	37.02
0.25	4	9.1	38.44
0.30	**3**	**4.6**	**38.75 (selected)**
0.40	3	3.8	38.68
0.50	2	3.1	38.61
0.70	1	2.4	37.90

The plateau from $\epsilon = 0.25$ to 0.50 (variance < 0.008) is the stable region on Actor; below $\epsilon \approx 0.20$ the outlier rate rises sharply and accuracy degrades. This pattern is consistent across all seven datasets.

A.14 End-to-End Training Cost and Inference Speedup

Struc2Vec is a one-time offline preprocessing step whose cost is amortised across all subsequent runs and seeds. Total MIDAS training cost is comparable to training three separate GNN teachers ($3\times$ single-teacher cost), with the Struc2Vec step paid only once. At inference, MIDAS accesses

Table 17: ϵ sweep on PubMed.

ϵ	# Clusters	Outlier %	Test Acc.
0.10	9	38.5	76.14
0.15	7	24.1	78.33
0.20	5	12.8	80.10
0.25	**4**	**6.3**	**81.70 (selected)**
0.30	4	5.7	81.58
0.40	3	4.9	81.41
0.50	2	3.8	81.20
0.70	1	2.6	80.05

only raw node features and executes a single trunk forward pass followed by one adapter — no adjacency matrix, no neighbourhood lookup — giving a 7–22× speedup over teacher GNNs and a 13–22× speedup over running all three teachers sequentially.

Table 18: Training cost (seconds, measured on A100).

Phase	Actor	PubMed	Co-CS	A-Photo
Avg. single teacher	47.0	88.7	371.5	105.5
Struc2Vec (one-time)	24.5	29.0	23.2	22.2
DBSCAN + center loss	0.9	0.9	0.6	0.3
Dual-encoder training	2.1	4.3	10.5	1.7
Distillation	11.2	24.9	16.5	6.2
Total MIDAS	**38.7**	**59.1**	**50.8**	**30.4**

Table 19: Inference latency (CPU, milliseconds) and speedup.

Method	Actor	PubMed	Co-CS	A-Photo
Avg. teacher	47.0	88.7	371.5	105.5
Sequential ensemble	141.7	266.1	1114.5	316.5
MIDAS student	**6.61**	**20.3**	**51.1**	**15.7**
Speedup vs. avg. teacher	7.1×	4.4×	7.3×	6.7×
Speedup vs. ensemble	21.4×	13.1×	21.8×	20.2×

A.15 Structural Descriptor Preprocessing

MIDAS computes structural descriptors using degree-aware random walks followed by Skip-Gram, as described in Sec. 4.1. This procedure is not Struc2Vec: it does not construct a multilayer structural-context graph or compute pairwise FastDTW distances. Accordingly, the cost of exact or κ-NN approximate Struc2Vec is not a preprocessing cost of MIDAS.

Let G' be the subgraph induced by $\mathcal{V}_{\text{train}} \cup \mathcal{V}_{\text{val}}$, with n' nodes and m' edges. Suppose we generate R walks of length H from each node. Computing degrees and the degree-aware transition weights takes $O(n' + m')$ time. If a sampling table is precomputed for each node's neighbor distribution, walk generation takes $O(n'RH)$ time and the tables require $O(m')$ storage. Without such tables, sampling a transition may instead require work proportional to the current node's degree.

The walks yield $O(n'RH)$ tokens for Skip-Gram training. Its additional cost depends on the context-window size, embedding dimension, number of negative samples, and training epochs. Walks from different source nodes can be generated independently. Structural descriptors are computed only for training and validation nodes during preprocessing; test-time prediction uses neither walks nor structural descriptors.

B NOTATION AND DEFINITIONS

This section fixes every symbol used in Secs. 3–4 and in the theoretical appendix, and states the definitions that the main text uses but does not display. Bold lower case denotes vectors, bold upper case matrices, calligraphic letters sets. $|\mathcal{Y}|$ and C both denote the number of classes ($C \equiv |\mathcal{Y}|$; the appendix uses C). Teacher output distributions are written $p_{\Theta_t^k}(\cdot \mid v)$ in the main text and $T^k(v)$ in the appendix; the two are identical.

B.1 SYMBOL TABLE

Table 20: Symbols used in the framework and the appendix.

Symbol	Meaning
Graph and data	
$G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$	Input graph: nodes, undirected edges, feature matrix $\mathbf{X} \in \mathbb{R}^{\lvert\mathcal{V}\rvert \times d}$
$\mathbf{x}_v \in \mathbb{R}^d$	Feature row of node v; the only input available at inference
$\mathcal{V}_{\text{train}}, \mathcal{V}_{\text{val}}, \mathcal{V}_{\text{test}}$	Node split; labels $y_v \in \mathcal{Y}$ are used only on $\mathcal{V}_{\text{train}}$ (and $\mathcal{V}_{\text{val}}$ for early stopping)
$\mathcal{N}(w)$, $\deg(w)$	Neighbour set and degree of w; in the inductive setting, both on the graph induced by $\mathcal{V}_{\text{train}} \cup \mathcal{V}_{\text{val}}$
$\mathbf{A}$, $\mathbf{D}$	Adjacency and degree matrices (RQ5 only)
h, h_{adj}	Edge homophily and adjusted homophily Platonov et al. (Table 8)
Teachers	
K	Number of teachers ($K = 3$: GCN, GAT, GRAPHSAGE)
Θ_t^k	Parameters of teacher k, frozen after pre-training
$p_{\Theta_t^k}(\cdot \mid v) \equiv T^k(v)$	Teacher k's predictive distribution at v (softened by T_{KD} where stated)
$\text{Acc}(\Theta, \mathcal{S})$	Accuracy of teacher Θ on node set $\mathcal{S}$
k_l^*	Teacher assigned to region l (Eq. 5)
$k^\star$	Globally most accurate teacher on $\mathcal{V}_{\text{train}}$ (Eq. 16)
$\mathcal{K}^*$, $\mathcal{K}$	Set of region-selected teachers $\{k_l^* : l \in \mathcal{R}\}$, and teachers in use $\mathcal{K} = \mathcal{K}^* \cup \{k^\star\}$
$\kappa(v) \in \{1, \dots, K\}$	Teacher supervising node v (Eq. 17)
Structural descriptor (Sec. 4.1)	
$\psi(v) \in \mathbb{R}^{c_s}$	Structural descriptor: Skip-Gram embedding of degree-aware walks; c_s is its dimension (App. B.4)
σ	Mean node degree; temperature of the walk transition $P(u \mid w) \propto e^{-\lvert \deg(w) - \deg(u)\rvert / \sigma}$
R_{w}, H_{w}	Walks per node and walk length (App. A.15)
Dual encoder (Sec. 4.1)	
$f_{\text{feat}} : \mathbb{R}^d \to \mathbb{R}^{c'}$	Feature encoder, kept at inference; $\mathbf{z}_v^{\text{feat}} = f_{\text{feat}}(\mathbf{x}_v)$
$f_{\text{struct}} : \mathbb{R}^{c_s} \to \mathbb{R}^{c'}$	Structural encoder, training only; $\mathbf{z}_v^{\text{struct}} = f_{\text{struct}}(\psi(v))$
c'	Shared latent dimension of both encoders
$\text{sim}(\mathbf{a}, \mathbf{b})$	Cosine similarity $\mathbf{a}^\top \mathbf{b} / (\lVert\mathbf{a}\rVert_2 \lVert\mathbf{b}\rVert_2)$
τ	InfoNCE temperature (Eq. 13)
N_b	Number of nodes in one alignment batch
$\mathcal{L}_{\text{align}}$	Cross-view InfoNCE loss (Eq. 13)
$\mathcal{L}_{\text{center}}$, λ_1	Center loss (Eq. **??**) and its weight
$\mathcal{L}_{\text{ref}}$, λ_2	Cosine-classifier cross-entropy on labelled nodes (Eq. 14) and its weight; $\mathbf{w}_y$ are the ℓ_2-normalized class vectors
E_0, E_1, E_2	Epochs of the three encoder stages (Eq. 15)
Regions and routing (Secs. 4.1–4.2)	
ϵ, m	DBSCAN neighbourhood radius and minimum points for a core point (Eq. 4)
$\pi(v) \in \{1, \dots, L\} \cup \{-1\}$	Region of a train/val node; -1 is DBSCAN noise; fixed after step (ii)
L	Number of regions returned by DBSCAN
$c_l = \{v : \pi(v) = l\}$	Node set of region l
$\mathbf{C}_l$	Region center: ℓ_2-normalized mean of $\{\mathbf{z}_v^{\text{feat}} : v \in c_l\}$
$\mathcal{V}_{\text{cl}} = \{v : \pi(v) \neq -1\}$	Clustered (non-noise) nodes

Symbol	Meaning
$n_{\min}$	Minimum training nodes for a region to take part in teacher selection (25)
$\mathcal{R} \subseteq \{1, \dots, L\}$	Retained regions: $\lvert\mathcal{V}_{\text{train}} \cap c_l\rvert \geq n_{\min}$
$\widehat{\pi}(v)$	Nearest-center placement of any node from $\mathbf{x}_v$ alone (Eq. 18)
Student (Sec. 4.2)	
g_θ, θ	Shared trunk MLP and its parameters; $\hat{\theta}$ denotes θ after warm-up (frozen)
a_{φ_k}, φ_k	Adapter for teacher $k \in \mathcal{K}$ and its parameters; disjoint across k
$f_s(\mathbf{x}_v) = a_{\varphi_{\kappa(v)}}(g_\theta(\mathbf{x}_v))$	Student prediction (Eq. 4.2); $p_s(\cdot \mid v)$ its softmax
$q_{\hat{\theta}}(v)$	Distribution of the temporary warm-up head on the frozen trunk (appendix only)
E_{s}, E_{sw}	Total distillation epochs and the warm-up/specialization switch epoch
γ_{ce}, γ_{kl}	Weights of $\mathcal{L}_{\text{CE}}$ and of the KL term in Eq. 6
T_{KD}	Distillation temperature; $\mathcal{L}_{\text{KL}}(v, \Theta) = T_{\text{KD}}^2 \,\text{KL}(p_\Theta(\cdot \mid v) \,\|\, p_s(\cdot \mid v))$
Analysis (Sec. 5)	
$c(v)$	Gradient conflict of v: minimum pairwise cosine of $\nabla_\theta \mathcal{L}_{\text{KL}}(v, \Theta_t^k)$ over teacher pairs, at the end of warm-up
$\mathbf{S}$	One-hot class-assignment matrix, $\mathbf{S} = \text{onehot}(\arg\max \hat{\mathbf{Y}})$ (RQ5); $\hat{\mathbf{Y}}$ the student's predicted distributions
Theory (App. A.6–A.9)	
$\bar{T}(v) = \frac{1}{K}\sum_k T^k(v)$	Average teacher
$T_k^{\text{rest}}(v) = \frac{1}{K-1}\sum_{j\neq k} T^j(v)$	Average of the other $K-1$ teachers
$T_l^*(v) = T^{k_l^*}(v)$	Assigned teacher of region l; T_*^{rest} its complement
$J(v)$	Teacher diversity $\frac{1}{K}\sum_k \text{KL}(T^k(v) \,\|\, \bar{T}(v))$
$R_1^{(l)}(\hat{\theta})$	Warm-up risk on region l: $\mathbb{E}_{v\sim c_l}[\text{KL}(\bar{T}(v) \,\|\, q_{\hat{\theta}}(v))]$
$\bar{d}_l$	$\mathbb{E}_{v\sim c_l}[d_{\text{TV}}(T_l^*(v), T_*^{\text{rest}}(v))]$
$d_{\text{TV}}(p,q) = \frac{1}{2}\lVert p - q\rVert_1$	Total variation distance
$R(T_k, c_l)$	Distillation risk of teacher k on region l (Prop. 1)
$\mathbf{z}_i^f, \mathbf{z}_i^s$	Shorthand for $\mathbf{z}_{v_i}^{\text{feat}}, \mathbf{z}_{v_i}^{\text{struct}}$, unit norm
$\bar{\mathbf{z}}^s, \tilde{\mathbf{z}}_i^s$	Batch mean of $\mathbf{z}^s$ and centered residual $\mathbf{z}_i^s - \bar{\mathbf{z}}^s$
$d_i, \bar{M}, \bar{S}, \bar{M}_\circ$	Cross-view gap $\lVert\mathbf{z}_i^f - \tilde{\mathbf{z}}_i^s\rVert_2$, its mean square, mean $\lVert\tilde{\mathbf{z}}_i^s\rVert_2^2$, and the ceiling of Cor. 1
$\mathbf{C}_l^s, \mathbf{C}_l^f$	Structural and feature centroids of region l (Thm. 1)
$\pi^s(i), \pi^f(i)$	Nearest-centroid routing in the structural and feature spaces
$n_l, w_{\min}$	Training nodes in region l; $\min_l n_l / N_{\text{tr}}$ with $N_{\text{tr}} = \lvert\mathcal{V}_{\text{train}}\rvert$
$\gamma_s, \varepsilon_{\text{c}}$	Structural routing margin and centroid-gap bound $\sqrt{\bar{M}/w_{\min}}$
$X_{i,k}, \mu_{l,k}, \hat{\mu}_{l,k}$	Correctness indicator, its mean and empirical mean for teacher k on region l (Prop. **??**)
$\hat{k}_l, \gamma_l$	Empirical argmax teacher of region l (this is Eq. 5's k_l^*) and the population margin

B.2 Definitions used in the main text without display

Alignment loss. For a batch $\{v_i\}_{i=1}^{N_b} \subseteq \mathcal{V}_{\text{train}} \cup \mathcal{V}_{\text{val}}$,

$$\mathcal{L}_{\text{align}} = -\frac{1}{N_b}\sum_{i=1}^{N_b} \log \frac{\exp\big(\text{sim}(\mathbf{z}_{v_i}^{\text{feat}}, \mathbf{z}_{v_i}^{\text{struct}})/\tau\big)}{\sum_{j=1}^{N_b} \exp\big(\text{sim}(\mathbf{z}_{v_i}^{\text{feat}}, \mathbf{z}_{v_j}^{\text{struct}})/\tau\big)}, \tag{13}$$

symmetrized over the two views. This is the InfoNCE objective of Oord et al. (2018).

Encoder cross-entropy (cosine classifier). With ℓ_2-normalized class vectors $\{\mathbf{w}_y\}_{y\in\mathcal{Y}}$ and the same τ,

$$\mathcal{L}_{\text{ref}} = -\frac{1}{\lvert\mathcal{V}_{\text{train}}\rvert}\sum_{v\in\mathcal{V}_{\text{train}}} \log \frac{\exp\big(\text{sim}(\mathbf{z}_v^{\text{feat}}, \mathbf{w}_{y_v})/\tau\big)}{\sum_{y\in\mathcal{Y}} \exp\big(\text{sim}(\mathbf{z}_v^{\text{feat}}, \mathbf{w}_y)/\tau\big)}. \tag{14}$$

Encoder schedule.

$$\underbrace{\mathcal{L}_{\text{align}}}_{E_0 \text{ epochs}} \xrightarrow{\text{DBSCAN (Eq. 4) fixes } \pi,\, L} \underbrace{\mathcal{L}_{\text{align}} + \lambda_1 \mathcal{L}_{\text{center}}}_{E_1 \text{ epochs}} \longrightarrow \underbrace{\mathcal{L}_{\text{align}} + \lambda_1 \mathcal{L}_{\text{center}} + \lambda_2 \mathcal{L}_{\text{ref}}}_{E_2 \text{ epochs}} \tag{15}$$

Centers $\mathbf{C}_l$ are recomputed from the current embeddings at every epoch; membership π is never changed after DBSCAN. f_{struct} and $\{\mathbf{w}_y\}$ are discarded afterwards.

Global teacher and teacher of a node.

$$k^\star = \operatorname*{argmax}_{k \in \{1,\dots,K\}} \operatorname{Acc}\big(\Theta_t^k, \mathcal{V}_{\text{train}}\big), \qquad \mathcal{K}^* = \{k_l^* : l \in \mathcal{R}\}, \qquad \mathcal{K} = \mathcal{K}^* \cup \{k^\star\}. \tag{16}$$

$$\kappa(v) = \begin{cases} k^*_{\pi(v)}, & \pi(v) \in \mathcal{R}, \\ k^\star, & \pi(v) = -1 \text{ or } \pi(v) \notin \mathcal{R}, \end{cases} \qquad \widehat{\kappa}(v) = k^*_{\widehat{\pi}(v)} \text{ for unseen } v. \tag{17}$$

Placement of unseen nodes.

$$\widehat{\pi}(v) = \operatorname*{argmax}_{l \in \mathcal{R}} \operatorname{sim}\big(f_{\text{feat}}(\mathbf{x}_v), \mathbf{C}_l\big) = \operatorname*{argmin}_{l \in \mathcal{R}} \big\|\text{normalize}(f_{\text{feat}}(\mathbf{x}_v)) - \mathbf{C}_l\big\|_2, \tag{18}$$

the two forms coinciding because both arguments are unit-norm. The same rule assigns training nodes to adapters during specialization, so train- and test-time routing are one operation.

B.3 Conventions that differ between the main text and the appendix

- Classes: $C \equiv |\mathcal{Y}|$. Teachers: $T^k(v) \equiv p_{\Theta_t^k}(\cdot \mid v)$; $T_l^* \equiv T^{k_l^*}$.
- Trunk: $g_{\hat{\theta}}$ (main text) is written $\hat{\psi}$ in the Rademacher argument; the adapter of teacher k is a_{φ_k} throughout, and in the Rademacher argument is restricted to the affine map $h \mapsto Wh + b$ with $\|W\|_{\text{F}} \le B_W$, $\|b\|_2 \le B_b$.
- $\kappa(v)$ is the teacher of a node; the Rademacher constant is written $\kappa_l^{\text{Rad}} = 2\sqrt{C}\,(B_W R_\psi + B_b)$ to avoid the clash.
- N_b is the alignment batch size; $N_{\text{tr}} = |\mathcal{V}_{\text{train}}|$ is the training-set size in Thm. 1; $|\mathcal{V}|$ is the node count in RQ5.
- ϵ is the DBSCAN radius; ε_{c} the centroid-gap bound. σ is the mean degree; σ_i, σ_{ic} are Rademacher signs. τ is the InfoNCE temperature; T_{KD} the distillation temperature.
- In Prop. **??**, $\hat{k}_l$ is the empirical selection of Eq. 5 and k_l^* there denotes the population optimum; elsewhere k_l^* is the empirical selection.

B.4 Hyperparameter Search Space

Table 21 lists the search space used to produce all results reported in the paper. Selection was performed via grid search on the validation set. An 80/10/10 train/validation/test split was used for datasets without pre-defined masks.

Table 21: Hyperparameter search space used for all reported results.

Hyperparameter	Search values
Hidden dim	256
Adapter bottleneck	64
Layers	$\{2, 3\}$
Temperature τ	$\{2, 4, 8\}$
Confidence threshold	$\{0.1, 0.3, 0.5\}$
λ_1 (center loss weight)	$\{0.001, 0.01, 0.1\}$
λ_2 (CE loss weight)	$\{0.5, 0.7, 1.0\}$
Optimizer	{Adam, SGD}
Activation	{Tanh, GELU}
Learning rate	$\{0.001, 0.005, 0.01\}$
Struc2Vec dim	$\{64, 128, 256\}$
DBSCAN ϵ	$\{0.2, 0.3, 0.4, 0.5\}$
DBSCAN MinPts	10

Table 22: `train_teachers.py`: defaults and tuning ranges.

Parameter	Default	Tuning range	Notes
`HIDDEN_DIM`	128	64–512	GNN hidden dimension for all teachers
`LR`	0.01	1e-3–5e-2	Adam learning rate
`WEIGHT_DECAY`	5e-4	1e-5–1e-3	ℓ_2 regularisation
`EPOCHS`	200	100–500	Max training epochs
`PATIENCE`	20	10–50	Early stopping patience
`DROPOUT_SAGE/GCN`	0.5	0.3–0.7	
`DROPOUT_GAT`	0.6	0.4–0.7	
`GAT_HEADS`	8	4–16	Attention heads for GAT
`NUM_SEEDS`	5	–	Number of independent runs

Table 23: `clustering.py`: defaults and tuning ranges.

Parameter	Default	Tuning range	Notes
`STRUC2VEC_DIM`	128	64–256	Embedding dim for Struc2Vec / dual-encoder output
`WALKS_PER_NODE`	25	10–50	More walks = richer structural signal, slower
`WALK_LENGTH`	20	10–40	
`CLIP_HIDDEN`	256	128–512	Dual encoder hidden dimension
`CLIP_OUT`	128	64–256	Shared latent dim; student input dimension
`CLIP_LR`	1e-4	5e-5–5e-4	
`CLIP_EPOCHS`	300	100–500	
`CLIP_TEMP`	0.07	0.05–0.2	InfoNCE temperature; lower = sharper alignment
`CENTER_LOSS_WEIGHT`	0.001	1e-4–1e-2	Weight of center loss vs. InfoNCE + CE
`CENTER_LR`	0.05	0.01–0.5	SGD learning rate for cluster centers
`DBSCAN_EPS`	0.7	0.2–0.7	Most dataset-sensitive parameter (§A.13)
`DBSCAN_MIN_SAMPLES`	5	3–20	Min. cluster size; increase to reduce noise clusters

Table 24: `train_student.py`: defaults and tuning ranges.

Parameter	Default	Tuning range	Notes
`TRUNK_HIDDEN`	512	256–1024	Shared trunk width
`ADAPTER_HIDDEN`	128	64–256	Per-cluster adapter bottleneck; $\sim$0.1–0.3$\times$ trunk
`DROPOUT`	0.3	0.1–0.5	Applied in trunk and adapters
`LR`	5e-2	1e-3–1e-1	Phase 2 learning rate
`WEIGHT_DECAY`	5e-4	1e-5–1e-3	
`WARMUP_EPOCHS`	50	20–100	Phase 1 length; increase for larger graphs
`TOTAL_EPOCHS`	350	200–600	Phase 1 + Phase 2 combined budget
`PATIENCE`	30	20–50	Early stopping patience (Phase 2)
`KL_WEIGHT`	0.7	0.5–0.9	Weight on KL distillation loss
`CE_WEIGHT`	0.3	0.1–0.5	Weight on cross-entropy loss; need not sum to 1 with KL
`TEMPERATURE`	2.0	1.0–5.0	Distillation temperature; higher = softer teacher targets
`NUM_SEEDS`	1	–	Set to 10 to reproduce paper mean $\pm$ std