SIMPLEX-GT: NODE-TO-CLUSTER ATTENTION FOR GRAPH LEARNING UNDER MIXED HOMOPHILY AND HETEROPHILY

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world graphs frequently exhibit mixed homophily and heterophily, where local message passing and global attention provide complementary but often conflicting signals. Existing graph transformers (GTs) and GNN–GT hybrids combine these signals at a single level of abstraction, leading to interference and unstable training. We propose **SimPlex-GT**, which decouples the two by routing global reasoning through a small set of learned semantic prototypes rather than full node-to-node attention. Our **node-to-cluster (N2C) attention** computes prototypes per forward pass as a differentiable function of input features—distinguishing it from prior tokenization-based GTs that rely on free-parameter latents or codebook-style update rules—and replaces $\mathcal{O}(N^2d)$ pairwise attention with *exact* attention against $M \ll N$ prototypes at $\mathcal{O}(NMd)$ cost. A lightweight GCN branch supplies local inductive bias, and three regularizers stabilize the global branch: complementary filtering of input features, cluster smoothing over the coarse graph induced by soft assignments, and a teacher–student self-supervised objective that prevents prototype collapse. The model is pretrained without labels and evaluated with a frozen linear probe. Across 14 node-classification, 3 graph-classification, and 4 node-clustering benchmarks (17 datasets in total), SimPlex-GT improves over the strongest baseline in our comparison by **+5.95** points on Squirrel-filtered, **+0.69** points on Penn94, and **+2.45** points on Amazon-Ratings, matches or improves on homophilic Cora/CiteSeer/PubMed, and trains faster in total wall-clock time than NodeFormer (**2.3**×, from faster convergence at comparable per-epoch cost) and GREET (up to **8.5**×). The results suggest that the bottleneck in scalable graph attention is *what to attend to*, not how cheaply to approximate full attention. *Code will be released upon acceptance.*

1 INTRODUCTION

Graph-structured data underpin applications from social networks and recommender systems to molecular property prediction and transportation (Kipf & Welling, 2016a; Veličković et al., 2017; Wu et al., 2022b; Fout et al., 2017; Zhang et al., 2024). Two architectural families dominate node representation learning on graphs: Graph Neural Networks (GNNs) (Kipf & Welling, 2016a; Veličković et al., 2017; Hamilton, 2020), which propagate information through local neighborhoods, and Graph Transformers (GTs) (Dwivedi & Bresson, 2020; Rampášek et al., 2022), which attend over all node pairs. The two carry opposing inductive biases: GNNs excel on *homophilic* graphs where neighbors share labels but suffer from over-smoothing and over-squashing under long-range dependencies, while GTs handle *heterophilic* graphs and distant relationships well but lack local structural priors and scale as $\mathcal{O}(N^2d)$.

The mixed-regime challenge. Real-world graphs rarely fall cleanly into either regime: most exhibit *mixed homophily and heterophily*, where local and global signals are simultaneously informative but often *conflicting* (Platonov et al., 2023; Luan et al., 2024). Existing hybrid GNN–GT architectures combine the two within a single representation space, implicitly assuming local and global signals can be jointly modeled at the same level of abstraction. We argue this is the central failure mode: the issue is not insufficient modeling capacity, but *improper coupling*. When local aggregation and global attention operate at the same granularity over the same features, their updates interfere—local

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.

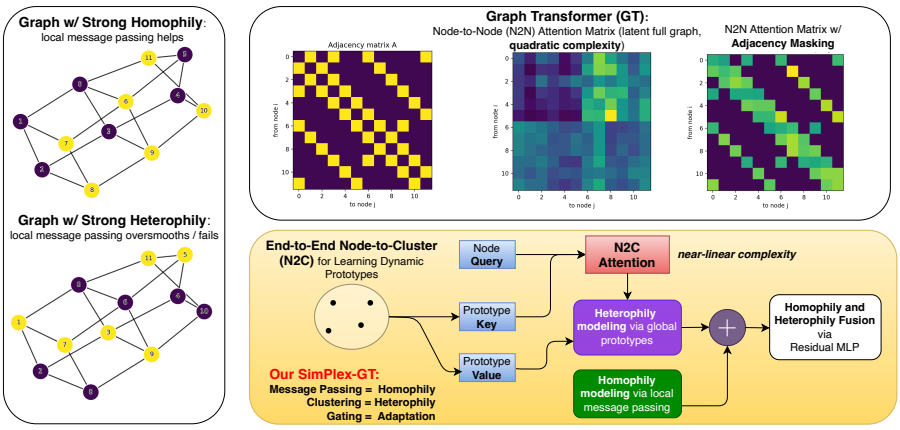


Figure 1: **Overview of SimPlex-GT.** We decouple global reasoning (via node-to-cluster attention over learned prototypes) from local message passing (via a lightweight GCN branch), and additively fuse them followed by a residual MLP.

smoothing corrupts long-range semantics, and global attention dilutes local structure—producing the training instability and inconsistent cross-regime performance that hybrid models routinely exhibit.

Aggregation before interaction. Our key insight is that local and global reasoning should operate at *different levels of abstraction*. For global reasoning we first aggregate nodes into a small set of learned semantic prototypes and then let every node attend to them: **node-to-cluster (N2C) attention**. Unlike inducing points, codebooks, or graph pooling (Sec. 2), N2C defines prototypes as a *differentiable function of the current node features* (Eq. 1), recomputed on every forward pass, so structural and statistical regularizers can shape them directly without any auxiliary update rule.

Our SimPlex-GT. SimPlex-GT (Fig. 1) has two branches. The *global branch* performs N2C attention over $M \ll N$ prototypes at $\mathcal{O}(NMd)$ instead of $\mathcal{O}(N^2 d)$ cost, stabilized by *complementary filtering* (CF), which builds prototypes from low-pass features and queries from the high-pass residual, or by *cluster smoothing* (CS) over the coarse graph induced by the soft assignments (Fig. 2). The *local branch* is a lightweight GCN. A teacher–student self-supervised objective with discrepancy-guided masking pretrains the whole model without labels and prevents prototype collapse; we evaluate with a frozen linear probe. The decoupling lies in how each branch forms its message: the global branch routes every node through the $N \to M \to N$ bottleneck and never sees individual neighbors, while the local branch sees only neighbors. Unlike GraphGPS-style hybrids, whose global path also attends node to node, the two paths operate at different granularities up to their additive fusion, where an orthogonality regularizer discourages redundancy.

Contributions. Our contributions are: **(i) Node-to-cluster attention.** Prototypes are a differentiable, per-forward-pass function of the input features, unlike the static or codebook latents of prior tokenization methods; this is what lets CF, CS, and a self-supervised objective act on prototype construction. **(ii) Decoupled local–global architecture.** We identify improper coupling—not insufficient capacity—as the core failure mode of hybrid GNN–GT models under mixed homophily, and pair N2C with a GCN branch that works at a different granularity. **(iii) The bottleneck is what to attend to, not how to approximate.** On the heterophilic and mixed-pattern benchmarks we evaluate, SimPlex-GT outperforms the kernelized, spectral, and tokenization-based scalable graph transformers in our comparison while using exact attention against $M \ll N$ prototypes, and replacing its input-conditioned prototypes with freely learned vectors degrades accuracy on every graph we test (Sec. 4.2).

2 Related Work

Scalable graph transformers. Sparse and hybrid designs restrict which node pairs interact (GraphGPS (Rampášek et al., 2022), Exphormer (Shirzad et al., 2023)); kernelized and linearized designs approximate the full softmax (NodeFormer (Wu et al., 2022a), SGFormer (Wu et al., 2023), Polynormer (Deng et al., 2024)). Both keep the N nodes as attention targets; N2C changes the targets to $M \ll N$ prototypes and attends to them exactly.

Aggregation before interaction. Set Transformer (Lee et al., 2019) and Perceiver (Jaegle et al., 2021) attend through free-parameter latents, NAGphormer (Chen et al., 2022b) tokenizes multi-hop neighborhoods, and GOAT (Kong et al., 2023) attends to a codebook updated by online K-means. Differentiable pooling (DiffPool (Ying et al., 2018), MinCutPool (Bianchi et al., 2020)) learns soft assignments S and coarsens $S^\top X$, $S^\top AS$ to downsample the graph for graph-level readout. N2C shares the soft assignments with pooling and the $N \to M$ bottleneck with inducing points, but uses the pooled, per-forward-pass prototypes as keys and values of node-level cross-attention, which lets CF, CS, and the SSL objective act on prototype construction directly.

Heterophilic and self-supervised graph learning. Heterophily-oriented GNNs redesign propagation (H2GCN (Zhu et al., 2020a), GPR-GNN (Chien et al., 2020)), keep high-frequency components (FAGCN (Bo et al., 2021), ACM-GCN (Luan et al., 2021)), or learn from global structure (LINKX (Lim et al., 2021), GloGNN (Li et al., 2022)); we adopt the curated benchmarks of Platonov et al. (2023). These methods combine local and global signals in one node-level representation, whereas SimPlex-GT keeps the global signal at prototype granularity until fusion. Self-supervised graph methods—contrastive (DGI (Veličković et al., 2019), GRACE (Zhu et al., 2020b)), reconstructive (GraphMAE (Hou et al., 2022)), negative-free (BGRL (Thakoor et al., 2021)), and heterophily-aware (MUSE (Yuan et al., 2023), GREET (Liu et al., 2022))—learn label-free representations; our objective follows BYOL/BGRL (Grill et al., 2020; Thakoor et al., 2021), is not claimed as a contribution, and is evaluated with the same frozen linear probe. App. C gives the extended discussion.

3 METHOD

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with $N = |\mathcal{V}|$ nodes, adjacency $A \in \{0,1\}^{N\times N}$, and node feature matrix $X \in \mathbb{R}^{N\times d}$. Each node v carries a label y_v. The (edge) homophily ratio is $\rho = |\{(u,v) \in \mathcal{E} : y_u = y_v\}|/|\mathcal{E}|$, with $\rho \to 1$ indicating homophily and $\rho \to 0$ heterophily.

3.1 NODE-TO-CLUSTER ATTENTION

Why global attention helps under heterophily. On heterophilic graphs, neighbors of v frequently carry labels different from y_v, so any local averaging contracts v's representation toward an uninformative or actively misleading mean. Global attention sidesteps this: a node attends to other nodes by feature similarity, not graph proximity. Theorem F.1 (App. F) makes this concrete in an idealized noiseless model with label-aligned features: one-layer mean aggregation misclassifies every node whose neighbors are mostly of the other class, while a single attention layer can classify every node correctly. It is an existence result, not a learning guarantee; Table 1 below validates it empirically and exposes its homophilic blind spot.

Limitations of full N2N attention. Standard graph transformers compute attention over all $N \times N$ node pairs. This has two costs. *Computationally*, memory and time scale as $\mathcal{O}(N^2)$, prohibitive beyond $\sim 10^4$ nodes. *Statistically*, every pairwise score $\langle q_i, k_j\rangle$ inherits noise from individual node features, so attention logits are high-variance and softmax weighting becomes brittle.

Aggregation before interaction. We replace pairwise interactions with interactions between nodes and a small set of learned prototypes. Given features $X \in \mathbb{R}^{N\times d}$, we compute $M \ll N$ prototypes via a learnable soft assignment:

$$C = \mathrm{Softmax}\big(XW_{\text{cluster}}\big) \in \mathbb{R}^{N\times M}, \quad Z = D_C^{-1} C^\top X \in \mathbb{R}^{M\times d}, \quad D_C = \mathrm{Diag}(C^\top \mathbf{1}_N). \tag{1}$$

Each prototype z_c is a (soft) weighted mean of node features, where row i of C gives node i's assignment distribution over the M prototypes, and the diagonal matrix D_C holds the cluster masses $s_c = \sum_i C_{ic}$, so that $z_c = \sum_i C_{ic} x_i / s_c$. The normalization is what makes z_c a mean rather than a sum: without it, prototype norms would grow with cluster mass ($\approx N/M$) and inflate the attention logits on large graphs. Theorem 1 is stated for exactly this normalized prototype. Attention then keeps queries at the node level but draws keys and values from prototypes:

$$\mathrm{N2C}(X) = \mathrm{Softmax}\Big(\frac{(XW_Q)(ZW_K)^\top}{\sqrt{d}}\Big)(ZW_V) \ \in \ \mathbb{R}^{N\times d}, \tag{2}$$

with computational cost $\mathcal{O}(NMd)$. Since M is a small constant (typically $5-15$, see Sec. 4), this is effectively linear in N.

Statistical effect of clustering. Beyond efficiency, clustering before attention reduces the variance of attention logits. The intuition is standard: if a prototype is a soft mean of n_{eff} effective members,

Table 1: **The heterophilic success & homophilic blind spot of global attention.** Node classification accuracy (across 10 splits) under supervised learning. OOM stands for "Out of Memory".

Methods	Heterophilic				Homophilic		
	Cornell	Texas	Wisconsin	Actor	Cora	CiteSeer	PubMed
GCN	57.03±3.30	60.00±4.80	56.47±6.55	30.83±0.77	81.50±0.30	70.30±0.27	79.00±0.05
GAT	59.46±3.63	61.62±3.78	54.71±6.87	28.06±1.48	83.02±0.19	72.51±0.22	79.87±0.03
MLP	81.08±7.93	81.62±5.51	84.31±3.40	35.66±0.94	56.11±0.34	56.91±0.42	71.35±0.05
GT-N2N	81.35±2.55	84.59±4.20	87.43 ±3.21	OOM	66.80±0.12	65.20±0.32	OOM
GT-N2C	82.16±1.79	87.60±4.26	88.23±3.18	36.13±0.62	73.16±0.18	68.12±0.30	74.90±0.08

its noise component is attenuated by roughly $1/n_{\text{eff}}$ relative to a single node's. The following result makes this precise.

Theorem 1 (Variance reduction, N2C vs. N2N; proof in App. G)**.** *Let* $x_i^{(0)} = y_i\mathbf{e} + \varepsilon_i$ *with* $\mathbb{E}[\varepsilon_i] = 0$ *and* $\mathrm{Cov}(\varepsilon_i) \preceq \sigma^2 I$*, and let prototypes* $\mathbf{p}_c = \frac{\sum_i p_{i,c} x_i^{(0)}}{\sum_i p_{i,c}}$ *be formed under soft assignments* $p_{i,c} \in [0,1]$ *with* $\sum_c p_{i,c} = 1$*. For attention output* $h_i^{N2C} = \sum_c \alpha_{i\to c}\mathbf{p}_c$*,*

$$\mathrm{Var}\big(\langle h_i^{N2C}, \mathbf{e}\rangle\big) \;\le\; \sigma^2\|\mathbf{e}\|^2 \sum_c \alpha_{i\to c} r_c \;\le\; \sigma^2\|\mathbf{e}\|^2, \quad \textit{with} \quad r_c = \frac{\sum_i p_{i,c}^2}{(\sum_i p_{i,c})^2} \le 1.$$

The bound for a single-node readout (N2N attention concentrated on one node), $\sigma^2\|\mathbf{e}\|^2$*, is recovered when* $r_c = 1$ *(singleton clusters).*

The factor r_c is the inverse effective cluster size, so larger, well-used clusters denoise more, whatever the attention distribution. Because diffuse N2N attention also averages after the softmax, the sharper distinction is at the logit level (Corollary G.1): N2C keys are denoised *before* the softmax, whereas each N2N logit carries a single node's noise.

Scope of the theory. Theorem 1 and the results we state in the appendix (Theorems F.1, H.1, and I.1) are conditional statements about idealized models. None shows that the learned prototypes are good, that N2C is more expressive than full attention, or that SimPlex-GT improves classification; they explain why the design choices are reasonable, and Sec. 4 tests whether they pay off.

The homophilic blind spot. Table 1 confirms both halves of the argument. GT-N2N and GT-N2C beat classical GNNs on Cornell, Texas, Wisconsin, and Actor by 4–27 points, and N2C avoids the OOM that N2N hits on Actor and PubMed; on Cora, CiteSeer, and PubMed both lose 5–10 points to GCN/GAT, because a purely global mechanism discards informative neighborhood structure. **Global attention alone is insufficient when local structure carries signal.**

3.2 Stabilizing Prototypes: CF-N2C and CS-N2C

The N2C mechanism leaves two issues: noisy features yield unstable assignments, and prototypes are built without regard to graph structure. We propose one N2C variant for each (Fig. 2).

i) CF-N2C: complementary filtering of features. Stable prototypes should reflect shared structure; queries should preserve discriminative detail. We separate these roles via a low-pass / high-pass decomposition of the feature matrix:

$$H_b \;=\; \hat{A}\,X, \quad H_f \;=\; X - H_b, \quad \hat{A} = \tilde{D}^{-1/2}(A+I)\tilde{D}^{-1/2}, \tag{3}$$

where $\tilde{D}$ is the degree matrix of $A+I$; H_b is a graph low-pass signal and H_f its high-pass complement. CF-N2C uses H_b for assignment and prototypes, and H_f for queries:

$$C = \mathrm{Softmax}(H_b W_{\text{cluster}}), \quad Z = D_C^{-1} C^\top H_b, \quad Q = H_f W_Q, \quad K = ZW_K, \quad V = ZW_V. \tag{4}$$

The intuition is bias-variance: prototypes inherit the smoothness (low variance) of H_b while queries retain the discriminative power (low bias) of H_f. Theorem H.1 (App. H) makes this precise: under a signal-plus-noise model, the CF logit has lower mean-squared error against its noise-free target than the unfiltered logit, for the noise lying in the non-negative spectrum of $\hat{A}$.

ii) CS-N2C: smoothing prototypes over the coarse graph. The second design issue is that prototypes computed via Eq. 1 are constructed independently—two prototypes that summarize structurally adjacent regions of the graph have no mechanism to share information. CS-N2C addresses this by lifting the input adjacency through the soft assignment to form a *coarse graph* on the prototypes

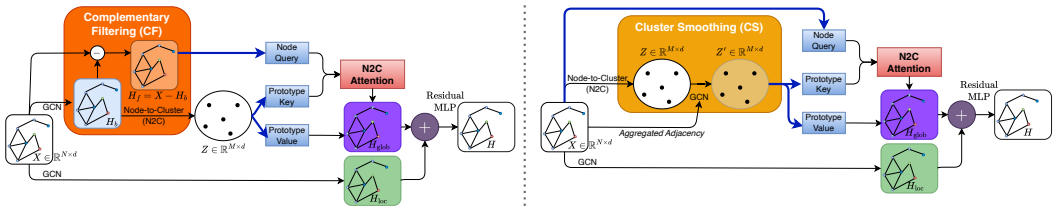


Figure 2: **Two variants of N2C attention.** *Left:* CF-N2C separates smoothed (low-pass) features for prototype construction from residual (high-pass) features for queries. *Right:* CS-N2C smooths prototypes over the coarse graph induced by soft assignments.

themselves, then smoothing over it:

$$A_c = C^\top A\, C - \operatorname{diag}(C^\top A\, C) \in \mathbb{R}^{M\times M}, \qquad Z' = (1-\alpha)\, Z + \alpha\, \hat{A}_c\, Z, \tag{5}$$

where the diagonal subtraction zeros out self-loops in the coarse graph (mass $C^\top AC$ would otherwise place on diag from same-cluster edges, double-counting against the residual $(1-\alpha)Z$), $\hat{A}_c$ is the symmetric-normalized A_c, and $\alpha \in (0,1)$ controls smoothing strength. Keys and values are drawn from Z' rather than Z. The construction inherits the standard residual GCN intuition—smoothing reduces prototype variance without forcing all prototypes toward a single mean—now applied at the cluster level rather than the node level.

Theorem I.1 (App. I) shows that one smoothing step shrinks prototype variance by the factor $(1-\alpha)^2 + \alpha^2 < 1$ when assignments are close to hard, at the cost of a bias toward neighboring prototypes that α controls.

When does each variant help? CF acts on inputs, denoising features before clustering; CS acts on outputs, sharing information across structurally related prototypes. Empirically the two differ by at most 1.1 points on most datasets (Sec. 4), so we report both as a robustness check on prototype stabilization; combining them did not help.

3.3 Two-Branch Fusion and Orthogonality Regularization

The N2C branch supplies global semantics; we add a parallel local branch to recover the structural signal that pure global attention discards. The local branch is a single GCN layer:

$$H_{\text{loc}} = \sigma\big(\hat{A}\, X\, W_{\text{loc}}\big), \tag{6}$$

applied in parallel to the global output $H_{\text{glob}} = \text{N2C}(X)$ from either CF or CS variant. The two outputs are summed and refined by a feedforward block:

$$H' = H_{\text{loc}} + H_{\text{glob}}, \qquad H = \text{LayerNorm}\big(H' + \text{MLP}(H')\big). \tag{7}$$

We use simple addition rather than a learned gate: addition keeps both branches active at every node, with relative influence determined by the magnitude each branch produces—branches whose output is uninformative at a given node tend to produce smaller magnitudes there, so addition realizes a soft, magnitude-driven selection without any learned gating parameters. This matters on graphs where homophily varies across nodes; on Tolokers, for example, 32.8% of nodes are strongly homophilic and 15.5% are heterophilic (App. O), and a fixed mixing weight would underserve at least one regime.

Orthogonality regularization. Without explicit pressure, the two branches can converge toward representations that encode the same information up to sign, defeating the decoupling rationale. We discourage redundancy with a per-node cosine penalty:

$$\mathcal{R}_\perp = \frac{1}{N} \sum_{v\in\mathcal{V}} \big|\cos\big(H_{\text{loc}}(v),\, H_{\text{glob}}(v)\big)\big|. \tag{8}$$

The absolute value penalizes both redundancy ($\cos \to 1$) and mirror representations ($\cos \to -1$); we found this preferable to penalizing only positive correlation, since branches that converge to negated versions of each other carry no more independent information than redundant ones. The regularizer is mild ($\lambda_{\text{orth}} \in [0.3, 0.5]$ in practice; App. M) and acts as a tiebreaker—when the matching loss is otherwise indifferent between two solutions, $\mathcal{R}_\perp$ pushes toward the one that uses the two branches for distinct information. A single fusion block suffices in our experiments.

3.4 Self-Supervised Training for Stable Prototype Learning

Prototype learning needs a training signal that reaches the cluster assignments. Without one, C tends to either *collapse* (most nodes routed to one prototype) or *drift* (prototype identities permute across

epochs without converging). We address this with a teacher–student self-supervised objective adapted from BYOL (Grill et al., 2020) and its graph counterpart BGRL (Thakoor et al., 2021), which serves as the label-free pretraining objective of SimPlex-GT. Its gradient flows through the soft assignments, so the same objective that trains the encoder also shapes prototype formation. The objective itself is not our contribution; the architecture it trains is.

Teacher–student framework. A student network $S(\cdot;\phi)$ processes a partially masked view of $\mathcal{G}$, while a teacher $T(\cdot;\psi)$ processes the full graph. Both share the SimPlex-GT architecture; teacher parameters ψ are maintained as an exponential moving average (EMA) of ϕ and receive no gradient updates. The student is trained to match the teacher in latent space:

$$\mathcal{L}(\phi) \;=\; \frac{1}{N}\sum_{v\in\mathcal{V}} \big\|S(v;\phi) - T(v;\psi)\big\|_2^2 \;+\; \lambda_{\text{orth}}\,\mathcal{R}_{\perp}(\phi), \tag{9}$$

where $\mathcal{R}_{\perp}$ is the same orthogonality regularizer introduced in Eq. 8.

Training and evaluation protocol. We pretrain the student encoder—both branches, the prototype assignment, and the fusion block—with Eq. 9 alone, using no labels. We then freeze it and train a linear classifier with cross-entropy on its output node representations, using the labeled training nodes of each split. This is the same frozen linear-probe protocol we apply to every self-supervised baseline (Sec. 4), and labels never update the encoder. The matching term is computed over *all* nodes, not just masked ones: visible nodes anchor representations against teacher drift, while masked nodes drive the predictive signal that backpropagates into the cluster assignments.

Discrepancy-guided masking. Random masking spends capacity on nodes the model already reconstructs well. We instead bias masking toward nodes the student currently disagrees with the teacher on. Define the per-node discrepancy $\tau(v) = \|S(v;\phi) - T(v;\psi)\|_2^2$ and let $\tau_{\max} = \max_{u\in\mathcal{V}} \tau(u)$. With overall masking ratio R and exploitation ratio $r \in [0,1]$, each node is independently masked with probability

$$p_v \;=\; \underbrace{(1-r)R}_{\text{exploration floor}} \;+\; \underbrace{rR\cdot \tau(v)/\tau_{\max}}_{\text{exploitation term}}, \tag{10}$$

which satisfies $p_v \in [(1-r)R,\, R]$ for all v. The exploration floor guarantees every node is masked with nontrivial probability, so no region of the graph becomes permanently visible (which would let assignments in that region drift unconstrained). The exploitation term concentrates effort on nodes that currently expose teacher–student disagreement. Training begins with pure random masking ($r=0$) and linearly ramps r to its target value over the first 10% of epochs. The discrepancies $\tau(v)$ and $\tau_{\max}$ are cached from the previous epoch's student and teacher outputs, so masking requires no extra forward pass; during the initial $r=0$ phase no discrepancies are needed. A masked node's features are replaced by a shared learnable mask token $\mathbf{m} \in \mathbb{R}^d$; the graph structure is left unchanged.

Why this stabilizes prototypes. We sketch the mechanism here; a full argument is in App. N. Suppose prototypes collapse, $z_1 = \cdots = z_M = z^\star$. Then the N2C context $c_v = \sum_k a_{v,k} z_k = z^\star$ is constant across nodes, so the student output $S(v;\phi) = f_\phi(z^\star)$ is constant too. The matching loss for masked nodes reduces to $\mathbb{E}_{v\in\mathcal{M}}\|\hat{y} - T(v;\psi)\|_2^2$ for some constant $\hat{y}$, whose minimum is $\mathrm{tr}(\mathrm{Cov}(T(v;\psi) \mid v \in \mathcal{M}))$. This lower bound is strictly positive whenever teacher targets vary across nodes—which they do at initialization (random projections produce diverse T) and continue to do under EMA updates. Collapse therefore incurs an irreducible loss penalty proportional to the variance of teacher targets, and the optimizer is pushed away from collapsed solutions. Empirically (App. N), prototype usage perplexity stays above 4 (out of $M=5$) under this objective and collapses below 2 within 50 epochs without it.

4 Experiments

We evaluate SimPlex-GT to answer three questions: **(Q1)** Does decoupling local and global reasoning yield consistent gains across homophily regimes? **(Q2)** Which design choices—N2C attention, CF/CS stabilization, the GCN branch, the SSL objective—are responsible for those gains? **(Q3)** How does N2C scale relative to existing scalable graph transformers?

Datasets. We evaluate on 17 benchmarks spanning three regimes. *Heterophilic:* Cornell, Texas, Wisconsin, Actor, and Chameleon-filtered, Squirrel-filtered, Roman-Empire (the Platonov et al. (Platonov et al., 2023) replacements for the small datasets). *Homophilic:* Cora, CiteSeer, PubMed, ogbn-Arxiv. *Mixed-pattern, large-scale:* Penn94, Tolokers, Amazon-Ratings (Platonov et al., 2023; Luan

Table 2: Node classification accuracy (%) reported as mean ± standard deviation across 10 data splits. The best, runner-up and third best results for each dataset are highlighted.

Category	Methods	Heterophilic Datasets				Homophilic Datasets			
		Cornell	Texas	Wisconsin	Actor	Cora	CiteSeer	PubMed	Arxiv
	Homo Ratio	0.30	0.11	0.21	0.22	0.81	0.74	0.80	0.66
	Semi-Supervised Learning (SL)								
GNNs	GCN (Kipf & Welling, 2016a)	57.03±3.30	60.00±4.80	56.47±6.55	30.83±0.77	81.50±0.30	70.30±0.27	79.00±0.05	71.74±0.27
GNNs	GAT (Veličković et al., 2017)	59.46±3.63	61.62±3.78	54.71±6.87	28.06±1.48	83.02±0.19	72.51±0.22	79.87±0.03	71.92±0.17
GNNs	MLP	81.08±7.93	81.62±5.51	84.31±3.40	35.66±0.94	56.11±0.34	56.91±0.42	71.35±0.05	55.50±0.23
GNNs	WRGAT (Suresh et al., 2021)	81.62±3.90	83.62±5.50	86.98±3.78	36.53±0.77	81.97±1.50	70.85±1.98	80.86±0.55	—
GNNs	H2GCN (Zhu et al., 2020a)	82.16±4.80	84.86±6.77	86.67±4.69	35.86±1.03	81.76±1.55	70.53±2.01	80.26±0.56	—
GTs	GraphGPS (Rampášek et al., 2022)	66.22±3.87	75.41±1.46	78.04±2.88	36.95±0.65	82.84±1.03	72.73±1.23	79.94±0.26	70.97±0.41
GTs	NAGphormer (Chen et al., 2022b)	56.22±8.08	63.51±6.53	62.55±6.22	34.33±0.94	82.12±1.18	71.47±1.30	79.73±0.28	70.13±0.55
GTs	Exphormer (Shirzad et al., 2023)	54.05±4.41	77.84±2.21	69.94±3.33	35.77±0.45	82.77±1.38	71.63±1.19	79.46±0.35	72.44±0.28
GTs	‡ NodeFormer (Wu et al., 2022a)	65.77±4.59	69.37±2.55	73.86±4.33	35.11±1.13	82.20±0.90	72.50±1.10	79.90±1.00	59.90±0.42
GTs	‡ SGFormer (Wu et al., 2023)	74.97±4.31	76.15±1.99	78.79±2.89	37.46±0.84	84.50±0.80	72.60±0.20	80.30±0.60	72.63±0.13
	Self-Supervised Learning (SSL)								
Homo	DGI (Veličković et al., 2019)	63.35±4.61	60.59±7.56	55.41±5.96	29.82±0.69	82.29±0.56	71.49±0.14	77.43±0.84	70.19±0.73
Homo	MVGRL (Hassani & Khasahmadi, 2020)	64.30±5.43	62.38±5.61	62.37±4.32	30.02±0.70	83.03±0.27	72.75±0.46	79.63±0.38	70.88±0.51
Homo	BGRL (Thakoor et al., 2021)	57.30±5.51	59.19±5.85	52.35±4.12	29.86±0.75	81.08±0.17	71.59±0.42	79.97±0.36	71.24±0.35
Homo	GRACE (Zhu et al., 2020b)	54.86±6.95	57.57±5.68	50.00±5.83	29.01±0.78	80.08±0.53	71.41±0.38	80.15±0.34	70.96±0.31
Homo	GraphMAE (Hou et al., 2022)	61.93±4.59	67.80±3.37	58.25±4.87	31.48±0.56	84.20±0.40	73.20±0.39	81.10±0.34	71.75±0.17
Hetero	DSSL (Xiao et al., 2022)	53.15±1.28	62.11±1.53	56.29±4.42	28.36±0.65	83.06±0.53	73.20±0.51	81.25±0.31	70.13±0.25
Hetero	HGRL (Chen et al., 2022a)	77.62±3.25	77.69±2.42	77.51±4.03	36.66±0.35	80.66±0.43	68.56±1.10	80.35±0.58	68.55±0.38
Hetero	* S3GCL (Wan et al., 2024)	81.27±3.67	86.12±3.91	84.56±2.71	36.88±0.34	*—	*—	*—	71.36±0.60
Hetero	†MUSE (Yuan et al., 2023)	82.00±3.42	83.98±2.81	88.24±3.19	36.15±1.21	82.22±0.21	71.14±0.40	82.90±0.40	70.98±0.32
Hetero	GREET (Liu et al., 2022)	73.51±3.15	83.80±2.91	82.94±5.69	35.79±1.04	83.84±0.71	73.25 ±1.14	80.29±1.00	71.09±0.43
Ours	SimPlex-GT -CF	83.78±4.68	92.97±4.39	91.37±2.35	37.51±0.87	84.75±0.18	73.36±0.16	83.25±0.20	72.44±0.20
Ours	SimPlex-GT -CS	84.86±3.24	92.97±3.66	92.16±2.63	37.68±0.83	84.02±0.16	73.12±0.18	81.42±0.32	71.97±0.32

‡ Original papers report limited heterophilic results, we supplement them with our reproduction and those from (Tang et al., 2025).

† MUSE provides hyperparameters only for Cornell, with unreproducible results; we tuned the others ourselves.

* As S3GCL's code is under construction, we report published results except on Cora, CiteSeer, and PubMed, which use different splits with higher label rates.

et al., 2024). We additionally evaluate graph classification on Peptides-func (Dwivedi et al., 2022), ogbg-molhiv, and ogbg-molpcba (Hu et al., 2020; 2021), and node clustering on four graphs (three heterophilic, plus CiteSeer) (App. D). We use the standard 10 splits of (Platonov et al., 2023) where applicable, 5 splits for Penn94/Amazon-Ratings per the official protocol, and the canonical 20-labels-per-class split for Cora/CiteSeer/PubMed. Full statistics in App. P.

Baselines. We compare against **(1) supervised GNNs** (GCN, GAT, MLP) and heterophily-oriented variants (WRGAT, H2GCN, GPR-GNN, FAGCN, MixHop, ACM-GCN, LINKX, GloGNN); **(2) graph transformers** (GraphGPS, NAGphormer, Exphormer, NodeFormer, SGFormer); and **(3) self-supervised methods**, both general (DGI, MVGRL, BGRL, GRACE, GraphMAE) and heterophily-aware (DSSL, HGRL, S3GCL, MUSE, GREET). For SSL methods we follow the standard linear-probe protocol (Yuan et al., 2023): encoders are frozen post-pretraining and a linear classifier is trained on labels. SimPlex-GT is evaluated under exactly this protocol (Sec. 3.4). Its encoder never sees labels, so its comparison with the supervised GNNs and GTs, which train end-to-end on labels, is conservative. We reproduce key baselines on identical splits where original numbers are unavailable. Baseline sources, tuning protocol, and hardware (which defines OOM) are given in the Reproducibility Statement.

4.1 Q1: Cross-Regime Performance

Table 2 reports node classification on the standard heterophilic and homophilic benchmarks; Table 3 on the harder Platonov benchmarks; Tables 4 and 5 on large mixed-pattern graphs.

Heterophilic graphs (the strong case). On the Platonov-curated heterophilic benchmarks, SimPlex-GT achieves its largest gains. On Squirrel-filtered, SimPlex-GT-CF reaches 47.75% versus 41.80% for the strongest baseline, SGFormer (+5.95 points; +6.18 over the strongest SSL baseline, MUSE); on Chameleon-filtered, 50.48% versus 46.48% (+4.00). On Roman-Empire, 81.56% trails GraphGPS (82.00%) by 0.44 points and exceeds every SSL baseline by at least 15.30 points. Cornell, Texas, and Wisconsin follow the same pattern with standard deviations of ±3–7 points on 183–251 nodes, so we do not emphasize them (Platonov et al., 2023).

Mixed-pattern large-scale graphs. On Penn94 (41K nodes) and Amazon-Ratings (24K nodes), SimPlex-GT improves over the strongest existing method by +0.69 points (GloGNN) and +2.45 points (GraphSAGE). The gains are modest but consistent: heterophily methods such as H2GCN and GloGNN collapse on Amazon-Ratings (36.47%, 36.89%), and homophily-oriented methods underperform on Squirrel.

Table 3: Accuracy (%) on three heterophilic graphs.

Methods	Chameleon(filtered)	Squirrel(filtered)	Roman-Empire
Homo Ratio	0.24	0.21	0.05
Semi-Supervised Learning (SL)			
GCN	40.89±4.12	39.47±1.47	73.69±0.74
GPR-GNN (Chien et al., 2020)	39.93±3.30	38.95±1.99	64.85±0.27
FAGCN (Bo et al., 2021)	41.90±2.72	41.08±2.27	65.22±0.56
H2GCN	26.75±3.64	35.10±1.15	60.11±0.52
GraphGPS	40.79±4.03	39.67±2.84	82.00±0.61
NodeFormer	34.73±4.14	38.52±1.57	64.49±0.73
SGFormer	44.93±3.91	41.80±2.27	79.10±0.32
Self-Supervised Learning (SSL)			
DGI	32.61±2.92	38.78±2.34	43.16±0.78
BGRL	32.55±4.65	35.67±1.42	52.16±0.25
GRACE	35.39±3.58	36.21±2.81	51.58±0.98
MUSE	46.48±2.51	41.57±1.44	66.26±0.53
GREET	44.67±2.98	39.69±1.85	63.37±1.91
SimPlex-GT -CF	50.32±3.20	47.75±1.65	81.56±0.68
SimPlex-GT -CS	50.48±3.22	47.56±1.63	79.88±0.78

Table 4: Accuracy (%) on **Penn94** (categorized as *amiguous heterophily*).

Model	Accuracy (%)
MLP	73.61 ± 0.40
GCN	82.47 ± 0.27
GAT	81.53 ± 0.55
GPRGNN	81.38 ± 0.16
MixHop	83.47 ± 0.71
ACM-GCN	82.52 ± 0.96
LINKX	84.71 ± 0.52
GloGNN	85.57 ± 0.35
SimPlex-GT-CF	86.26 ± 0.49
SimPlex-GT-CS	86.12 ± 0.66

Table 5: Accuracy (%) on large scale **Amazon-ratings.**

Model	Accuracy (%)
GCN	48.70 ± 0.63
GraphSAGE	53.63 ± 0.39
GAT	49.09 ± 0.63
H2GCN	36.47 ± 0.23
GPRGNN	44.88 ± 0.34
GloGNN	36.89 ± 0.14
FAGCN	44.12 ± 0.30
GT	51.17 ± 0.66
NodeFormer	43.78 ± 0.32
SGFormer	47.95 ± 0.42
GOAT	44.61 ± 0.50
SimPlex-GT-CF	55.53 ± 0.26
SimPlex-GT-CS	56.08 ± 0.32

Homophilic graphs (competitive, not dominant). On Cora, CiteSeer, PubMed, and ogbn-Arxiv, SimPlex-GT-CF is within 0.5 points of the best baseline and best on Cora and PubMed, differences within run-to-run variance (Platonov et al., 2023). Heterophily-focused methods such as HGRL lose 3–5 points here; SimPlex-GT does not, because its local GCN branch absorbs the homophilic signal.

Summary. SimPlex-GT is the only method in our comparison that is best-or-tied on most heterophilic graphs and within noise of the best on homophilic graphs. **This directly answers Q1:** decoupling produces gains where they matter without sacrificing performance where they don't.

4.2 Q2: Which Components Drive the Gains?

Table 6 ablates the four design choices on a representative subset of datasets. We disable each component while holding others fixed.

Table 6: Ablations of SimPlex-GT-CS. Each row changes one component and holds the rest fixed. "w/o N2C" replaces N2C with full N2N attention; "w/o GCN branch" removes H_{loc} from Eq. 7; "Free prototypes" replaces the input-conditioned prototypes Z with M freely learned vectors. Tolokers results are in App. J.

Methods	Cornell	Texas	Wisconsin	Actor	Cora	CiteSeer	PubMed	Arxiv
SimPlex-GT (CS)	84.86±3.24	92.97±3.66	92.16±2.63	37.68±0.83	84.02±0.16	73.12±0.18	81.42±0.32	71.97±0.32
w/o N2C	82.16±2.16	87.83±3.67	88.23±2.35	OOM	78.80±0.15	69.26±0.28	OOM	OOM
w/o CS	82.70±2.48	88.92±4.26	88.56±2.77	36.57±0.88	83.80±0.10	72.70±0.18	80.00±0.33	70.98±0.28
w/o SSL	83.24±3.91	90.54±4.05	90.39±3.21	37.07±1.10	80.60±0.16	72.10±0.21	78.00±0.32	71.88±0.36
w/o GCN branch	84.32 ±2.96	92.11 ±3.12	91.56 ±2.10	37.23 ±1.01	82.20 ±0.11	72.50±0.22	80.60±0.28	71.35±0.26
Free prototypes	81.08 ± 3.56	89.18 ± 4.12	88.64 ± 3.84	29.36 ± 1.10	78.30 ± 0.34	68.98 ± 0.34	78.50 ± 0.40	69.98 ± 0.42

N2C is the load-bearing component. Replacing N2C with full N2N attention causes either large drops (Cornell -2.7, Texas -5.1, Wisconsin -3.9) or out-of-memory failures (Actor, PubMed, Arxiv). This is consistent with Theorem 1: N2C's variance reduction matters most precisely when neighborhoods are noisy, which is the heterophilic regime where the drops are largest.

Input-conditioned prototypes carry the gain. We replace the input-conditioned prototype construction with M free prototype parameters while retaining the same downstream N2C attention, leads to significant performance drops across datasets. SimPlex-GT constructs prototypes as graph-conditioned weighted means, so each prototype is a data-supported, statistically aggregated representation of a semantic mode present in the graph. In contrast, freely learned prototypes are input-independent parameters optimized from scratch through the same training objective. Such a learned direction has no inherent connection to the semantic structure of feature space; whether it corresponds to an actual feature cluster must be discovered entirely through gradient-based optimization.

The local GCN branch matters most on homophilic graphs. Removing H_{loc} costs 1.82 points on homophilic Cora but 0.60 on heterophilic Wisconsin. The substantially larger degradation on homophilic Cora is consistent with the intended role of the local GCN branch: preserving informative neighborhood structure when homophily is strong.

Cluster smoothing matters most on heterophilic graphs. Removing CS from SimPlex-GT-CS costs 2.2–4.1 points on Cornell, Texas, and Wisconsin, 1.0–1.4 points on Actor, PubMed, and Arxiv, and 0.2–0.4 points on Cora and CiteSeer.

Table 7: Wall-clock time and peak GPU memory on two benchmarks (using SimPlex-GT-CF).

Datasets	GPU Memory(MB)				Epoch Time(s/epoch)				Total Time(s)			
	NodeFormer	MUSE	GREET	SimPlex-GT	NodeFormer	MUSE	GREET	SimPlex-GT	NodeFormer	MUSE	GREET	SimPlex-GT
Actor	**1157**	8785	4318	1793	0.08	0.45	0.58	**0.07**	25.20	53.09	66.71	**14.84**
Roman	**4048**	34798	36433	6863	0.23	2.48	2.82	**0.20**	103.04	301.35	379.21	**44.62**

SSL pretraining contributes, but is not the whole gain. Replacing SSL pretraining and the linear probe with end-to-end supervised training ("w/o SSL") costs 3.4 points on Cora and PubMed, 1.0 on CiteSeer, 0.1 on Arxiv, and 0.6–2.4 on the heterophilic graphs; with SSL the prototypes stay diverse and in use (App. N). Even without SSL, SimPlex-GT exceeds the N2N variant on every dataset where both run, and Table 1 compares N2N and N2C under purely supervised training.

Other components. CF and CS differ by at most 1.1 points on most dataset, where CF is ahead; neither variant dominates, so we report both as a robustness check. Removing the orthogonality regularizer costs 0.6–1.3 points (App. M); sensitivity to M and masking ratio r is in Apps. K and L.

4.3 Q3: Efficiency, and What "Scalable Attention" Actually Trades

Table 7 compares runtime and memory on Roman-Empire (22K nodes) and Actor (7.6K nodes), the two largest heterophilic benchmarks where every listed baseline runs without OOM.

Total training time is lower, mostly through faster convergence. On Roman-Empire, SimPlex-GT trains in 44.6 s: $2.3\times$ faster than NodeFormer, $6.8\times$ than MUSE, and $8.5\times$ than GREET ($1.7\times$, $3.6\times$, $4.5\times$ on Actor). Per-epoch time is comparable to NodeFormer's ($\sim$1.15$\times$); the gap comes from reaching early stopping in about half as many epochs ($\sim$223 vs. $\sim$448). The SSL baselines MUSE and GREET are 12–$14\times$ slower per epoch.

N2C is more accurate than the scalable graph transformers we evaluate on most graphs. On Roman-Empire, SimPlex-GT-CF reaches 81.56% versus NodeFormer 64.49% (kernelized) and SGFormer 79.10% (spectral), and trails the sparse/hybrid GraphGPS (82.00%) by 0.44 points. On Squirrel-filtered, the gap to all three exceeds around 6 points (Table 3). The pattern persists across mixed-pattern benchmarks (Penn94, Ratings).

N2C is exact, not approximate. The $\mathcal{O}(NMd)$ cost needs no random features, spectral truncation, sparsity heuristics, or auxiliary clustering updates. The explicit $N \times M$ assignment and attention matrices cost memory: 6.7 GB on Roman-Empire, more than NodeFormer but far below MUSE/GREET (35 GB), which we **trade for accuracy (+17.1 points) and total training time**. N2C stays linear in N only while M is fixed; App. A discusses when a fixed M should suffice and when it may fail.

Summary. On the heterophilic and mixed-pattern graphs we tested, SimPlex-GT is ahead of NodeFormer and the SSL baselines in both accuracy and total training time, and ahead of SGFormer in accuracy, while NodeFormer keeps a lower memory footprint. The bottleneck in scalable graph attention appears to be **what to attend to**, not how cheaply to approximate attention against all nodes.

4.4 Node Clustering and Graph Classification

To verify that the learned representations transfer beyond node classification, we evaluate on node clustering (k-means on frozen embeddings) and graph classification. Headline numbers: SimPlex-GT-CS beats MUSE by +12.33 on Cornell and +10.61 on Texas for clustering accuracy (App. D), and improves over GraphGPS by +6.77 on Peptides-func AP (App. E). Full tables and protocols are in the appendices. The pattern matches node classification: the largest gains arise on heterophilic and long-range datasets, where prototype-based global reasoning provides the most structural information.

5 Conclusion

We presented SimPlex-GT, which decouples local and global reasoning on graphs with mixed homophily and heterophily: a lightweight GCN branch handles local structure, and node-to-cluster (N2C) attention performs exact global attention against $M \ll N$ input-conditioned prototypes at $\mathcal{O}(NMd)$ cost, stabilized by complementary filtering or cluster smoothing and pretrained without labels. It achieves the strongest results in our comparison on the curated heterophilic and large mixed-pattern benchmarks, stays within noise of the best methods on homophilic graphs, and trains faster in total wall-clock time than NodeFormer and GREET. Replacing its input-conditioned prototypes with freely learned ones consistently degrades accuracy, which locates the gain in *what* the attention targets are.

REFERENCES

Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. Mixhop: Higher-order graph convolutional architectures via sparsified neighborhood mixing. In *international conference on machine learning*, pp. 21–29. PMLR, 2019.

Kristen M Altenburger and Johan Ugander. Monophily in social networks introduces similarity among friends-of-friends. *Nature human behaviour*, 2(4):284–290, 2018.

Jinheon Baek, Minki Kang, and Sung Ju Hwang. Accurate learning of graph representations with graph multiset pooling. In *International Conference on Learning Representations (ICLR)*, 2021.

Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 874–883, 2020.

Deyu Bo, Xiao Wang, Chuan Shi, Meiqi Zhu, Emiao Lu, and Peng Cui. Structural deep clustering network. In *Proceedings of the web conference 2020*, pp. 1400–1410, 2020.

Deyu Bo, Xiao Wang, Chuan Shi, and Huawei Shen. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 3950–3957, 2021.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9650–9660, 2021.

Jingfan Chen, Guanghui Zhu, Yifan Qi, Chunfeng Yuan, and Yihua Huang. Towards self-supervised learning on graphs with heterophily. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 201–211, 2022a.

Jinsong Chen, Kaiyuan Gao, Gaichao Li, and Kun He. Nagphormer: A tokenized graph transformer for node classification in large graphs. *arXiv preprint arXiv:2206.04910*, 2022b.

Eli Chien, Jianhao Peng, Pan Li, and Olgica Milenkovic. Adaptive universal generalized pagerank graph neural network. *arXiv preprint arXiv:2006.07988*, 2020.

Chenhui Deng, Zichao Yue, and Zhiru Zhang. Polynormer: Polynomial-expressive graph transformer in linear time. In *International Conference on Learning Representations (ICLR)*, 2024.

Vijay Prakash Dwivedi and Xavier Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Vijay Prakash Dwivedi, Ladislav Rampášek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. *Advances in Neural Information Processing Systems*, 35:22326–22340, 2022.

Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems*, 30, 2017.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 21271–21284, 2020.

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.

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

William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

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

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16000–16009, 2022.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Zhenyu Hou, Xiao Liu, Yukuo Cen, Yuxiao Dong, Hongxia Yang, Chunjie Wang, and Jie Tang. Graphmae: Self-supervised masked graph autoencoders. In *KDD*, 2022.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020.

Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*, 2021.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. Perceiver: General perception with iterative attention. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pp. 4651–4664. PMLR, 2021.

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

Thomas N. Kipf and Max Welling. Variational graph auto-encoders. In *NeurIPS Workshop on Bayesian Deep Learning*, 2016b.

Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Renkun Ni, C Bayan Bruss, and Tom Goldstein. Goat: A global transformer on large-scale graphs. In *International Conference on Machine Learning*, pp. 17375–17390. PMLR, 2023.

Devin Kreuzer, Dominique Beaini, William L. Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 21618–21629, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 3744–3753. PMLR, 2019.

Xiang Li, Renyu Zhu, Yao Cheng, Caihua Shan, Siqiang Luo, Dongsheng Li, and Weining Qian. Finding global homophily in graph neural networks when meeting heterophily. In *International Conference on Machine Learning*, pp. 13242–13256. PMLR, 2022.

Derek Lim, Felix Hohne, Xiuyu Li, Sijia Linda Huang, Vaishnavi Gupta, Omkar Bhalerao, and Ser-Nam Lim. Large scale learning on non-homophilous graphs: New benchmarks and strong simple methods. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 20887–20902, 2021.

Yixin Liu, Yizhen Zheng, Daokun Zhang, Vincent Lee, and Shirui Pan. Beyond smoothing: Unsupervised graph representation learning with edge heterophily discriminating. *arXiv preprint arXiv:2211.14065*, 2022.

Sitao Luan, Chenqing Hua, Qincheng Lu, Jiaqi Zhu, Mingde Zhao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Is heterophily a real nightmare for graph neural networks to do node classification? *arXiv preprint arXiv:2109.05641*, 2021.

Sitao Luan, Chenqing Hua, Qincheng Lu, Liheng Ma, Lirong Wu, Xinyu Wang, Minkai Xu, Xiao-Wen Chang, Doina Precup, Rex Ying, et al. The heterophilic graph learning handbook: Benchmarks, models, theoretical analysis, applications and challenges. *arXiv preprint arXiv:2407.09618*, 2024.

Yuankai Luo, Lei Shi, and Xiao-Ming Wu. Can classic gnns be strong baselines for graph-level tasks? simple architectures meet excellence. *arXiv preprint arXiv:2502.09263*, 2025.

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K. Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pp. 23321–23337. PMLR, 2023.

Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th international conference on World Wide Web*, pp. 201–210, 2007.

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.

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? *arXiv preprint arXiv:2302.11640*, 2023.

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

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.

Hamed Shirzad, Ameya Velingker, Balaji Venkatachalam, Danica J Sutherland, and Ali Kemal Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, pp. 31613–31632. PMLR, 2023.

Susheel Suresh, Vinith Budde, Jennifer Neville, Pan Li, and Jianzhu Ma. Breaking the limit of graph neural networks by improving the assortativity of graphs with local mixing patterns. *arXiv preprint arXiv:2106.06586*, 2021.

Jiachen Tang, Zhonghao Wang, Sirui Chen, Sheng Zhou, Jiawei Chen, and Jiajun Bu. Opengt: A comprehensive benchmark for graph transformers. *arXiv preprint arXiv:2506.04765*, 2025.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pp. 1067–1077, 2015.

Shantanu Thakoor, Corentin Tallec, Mohammad Gheshlaghi Azar, Mehdi Azabou, Eva L Dyer, Remi Munos, Petar Veličković, and Michal Valko. Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514*, 2021.

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

Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. Deep graph infomax. In *International Conference on Learning Representations*, 2019.

Guancheng Wan, Yijun Tian, Wenke Huang, Nitesh V Chawla, and Mang Ye. S3gcl: Spectral, swift, spatial graph contrastive learning. In *Forty-first International Conference on Machine Learning*, 2024.

Qitian Wu, Wentao Zhao, Zenan Li, David P Wipf, and Junchi Yan. Nodeformer: A scalable graph structure learning transformer for node classification. *Advances in Neural Information Processing Systems*, 35:27387–27401, 2022a.

Qitian Wu, Wentao Zhao, Chenxiao Yang, Hengrui Zhang, Fan Nie, Haitian Jiang, Yatao Bian, and Junchi Yan. Sgformer: Simplifying and empowering transformers for large-graph representations. *Advances in Neural Information Processing Systems*, 36:64753–64773, 2023.

Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022b.

Teng Xiao, Zhengyu Chen, Zhimeng Guo, Zeyang Zhuang, and Suhang Wang. Decoupled self-supervised learning for graphs. *Advances in Neural Information Processing Systems*, 35:620–634, 2022.

Rex Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Mengyi Yuan, Minjie Chen, and Xiang Li. Muse: Multi-view contrastive learning for heterophilic graphs. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 3094–3103, 2023.

Jiahao Zhang, Rui Xue, Wenqi Fan, Xin Xu, Qing Li, Jian Pei, and Xiaorui Liu. Linear-time graph neural networks for scalable recommendations. In *Proceedings of the ACM on Web Conference 2024*, pp. 3533–3544, 2024.

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.

REPRODUCIBILITY STATEMENT

Baselines. When a published result uses the same official split and protocol, we report it to avoid weakening a baseline through re-implementation; otherwise we run the official implementation on our splits, searching the hyperparameter space given in the original paper under the same tuning budget as SimPlex-GT. Table footnotes mark reproduced (‡), self-tuned (†), and published ($\star$) entries. All runs use full-graph training on a single NVIDIA A100; "OOM" means the run exceeds this maximum budget. Search spaces are listed in App. Q.

Section 3 specifies the model, including the normalized prototype construction (Eq. 1); App. Q lists the hyperparameter search spaces; App. P gives dataset statistics and splits; and the paragraph above states the baseline protocol and hardware. All proofs are in Apps. F–I. We will release code, per-dataset configurations with seeds, exact commands, and result logs for every table.

AI USE STATEMENT

We used LLMs solely to polish the writing (grammar, spelling, and word choice). They were not used to generate research ideas, methods, experiments, or results.

A LIMITATIONS

We discuss the scope of this work and directions for future research.

Fixed prototype count. The number of prototypes M is treated as a per-dataset hyperparameter (typically $M = 5$; see App. K for sensitivity). We do not learn M adaptively, nor do we let it vary across layers or graph regions. On our benchmarks, a small M suffices even with many classes: on Roman-Empire (18 classes), $M = 5$ reaches 80.86%, while increasing M to 15 provides only a further 0.7-point improvement (App. K). This is possible because soft assignment allows each node to interpolate across shared prototypes rather than requiring a one-to-one correspondence between prototypes and classes. Learning the prototype count adaptively or adopting layer-wise or hierarchical prototypes is an interesting direction for future work.

Scale of evaluation. The largest graph we evaluate is ogbn-Arxiv (170K nodes); we have not evaluated SimPlex-GT at the million-node scale, such as ogbn-Papers100M. Extending the empirical evaluation to substantially larger graphs is an important direction for future work.

Edge-level and dynamic graphs. Our experiments cover node classification, node clustering, and graph classification on static, undirected graphs. We have not evaluated SimPlex-GT on link prediction, heterogeneous (multi-relational) graphs, or temporal/dynamic graphs, where prototype construction may require extensions to accommodate typed or time-varying relations.

B BROADER IMPACTS

This work advances the methodology of graph representation learning, with downstream relevance to applications that already use graph neural networks: recommender systems, fraud and anomaly detection in transaction networks, drug- and protein-interaction modeling, social-network analysis, and traffic or infrastructure prediction. SimPlex-GT improves accuracy and efficiency on heterophilic and mixed-pattern graphs, which are common in social and financial networks where edges connect dissimilar entities (e.g., buyer–seller, follower–influencer, or heterophilic citation links). We see no application of the method that is uniquely enabled or amplified beyond what existing graph transformers and GNNs already make possible.

Potential positive impacts. Better cross-regime accuracy could improve the reliability of GNN-based decision-support tools in domains where heterophilic structure is prevalent (e.g., money-laundering detection on transaction graphs, drug-target interaction prediction on biomedical graphs). The near-linear $\mathcal{O}(NM)$ cost lowers the compute and energy footprint of training and inference relative to full graph transformers, which is a small but tangible step toward greener large-graph learning.

Potential negative impacts and risks. *Surveillance and discrimination on social graphs.* Like any graph-learning method, SimPlex-GT can be deployed on social or behavioral graphs in ways that profile individuals or amplify existing biases in the underlying data. The model itself is task-agnostic; the burden of fair use lies with deployers, who should evaluate disparate impact on protected subgroups before deployment. *Brittleness under distribution shift.* Our evaluation uses standard public benchmarks. Real deployments often face evolving graph distributions (new node types, drifting edge semantics) where prototype assignments learned on training data may no longer be meaningful. Practitioners should monitor prototype usage statistics (App. N) and re-train when drift is detected. *Adversarial robustness.* We do not characterize SimPlex-GT's behavior under adversarial structural perturbations (edge insertions/deletions or feature poisoning targeting prototype assignments). The soft-assignment matrix C is a plausible attack surface: an adversary who can perturb a small number of high-mass nodes could disproportionately influence a prototype. Robust variants of N2C are an open question. *Compute access.* While SimPlex-GT is more efficient than full graph transformers, training still benefits from GPU access; the gap between groups with and without compute access remains a structural concern for the field that this work does not address.

We did not collect any data, conduct any human-subjects work, or release any model trained on personal information. All experiments use widely available, previously published academic benchmarks (citation graphs, web graphs, social networks, molecular datasets) under their original licenses; dataset statistics and licenses are summarized in App. P.

C Related Work

C.1 Graph Transformers and Scalable Attention

Graph Transformers (GTs) extend self-attention to graph-structured data, enabling global interactions beyond the local message passing of standard GNNs (Dwivedi & Bresson, 2020; Rampášek et al., 2022; Kreuzer et al., 2021). Their central liability is the $\mathcal{O}(N^2)$ cost of full attention, which prior work has addressed along three lines.

Sparse and structured attention. GraphGPS (Rampášek et al., 2022) interleaves sparse local message passing with global attention and structural encodings. Exphormer (Shirzad et al., 2023) uses expander graphs to maintain global connectivity under fixed sparsity. GRIT (Ma et al., 2023) introduces relative random-walk encodings for tighter inductive bias. These methods retain pairwise attention but limit which pairs interact.

Kernel and spectral approximations. NodeFormer (Wu et al., 2022a) uses kernelized random-feature attention for near-linear scaling. SGFormer (Wu et al., 2023) combines a single-layer simplified attention with spectral sparsification. Polynormer (Deng et al., 2024) replaces softmax attention with polynomial expansions. These methods preserve full pairwise reach in expectation but trade exactness for scalability.

Tokenization and aggregation-before-interaction. A third line—most relevant to our work—compresses nodes into a smaller set of tokens before attention. NAGphormer (Chen et al., 2022b) tokenizes per-node neighborhoods into fixed-size sequences. GOAT (Kong et al., 2023) replaces N2N attention with attention against a learnable codebook, updated by an online K-means rule, achieving $\mathcal{O}(NM)$ cost. Set Transformer (Lee et al., 2019) establishes the general aggregation-before-interaction paradigm via *inducing points* that mediate cross-attention between input sets, and Perceiver (Jaegle et al., 2021) extends this to heterogeneous modalities. Set-Transformer-style designs have also been adapted for graph-level tasks (Baek et al., 2021). Differentiable graph pooling—DiffPool (Ying et al., 2018) and MinCutPool (Bianchi et al., 2020)—learns a soft assignment S with a neural network and forms the coarsened graph $S^\top X$, $S^\top AS$; it is the direct precursor of our soft assignment C and coarse graph $C^\top AC$ (Eq. 5). Pooling uses the coarse graph to *replace* the input graph on the way to a graph-level readout; N2C keeps all N nodes and uses the pooled prototypes only as keys and values of a node-to-cluster cross-attention.

Position of SimPlex-GT. Our N2C attention shares the aggregation-before-interaction skeleton with Set Transformer's inducing points and GOAT's learned codebook: in all three, attention is computed between N inputs and $M \ll N$ learned latents. The contribution we claim is not the attention mechanism in isolation but the system around it. Specifically, where prior tokenization methods treat

the latent set primarily as an *efficiency mechanism*—a bottleneck through which information must pass—we treat prototypes as a *structurally regularized representation* of the graph: their construction is conditioned on the input distribution via complementary filtering (CF), their interactions are shaped by graph topology via cluster smoothing (CS), and their stability is enforced by a teacher–student objective that prevents collapse. This shifts the role of clustering from compression to structured global abstraction.

C.2 Learning on Heterophilic Graphs

Heterophilic graphs arise in fraud detection (Pandit et al., 2007), social networks (Altenburger & Ugander, 2018), and molecular graphs (Zhu et al., 2020a), where connected nodes frequently differ in features or labels. Methods to handle heterophily fall into four broad categories.

Higher-order and rewired propagation. MixHop (Abu-El-Haija et al., 2019) and Geom-GCN (Pei et al., 2020) aggregate over multi-hop or latent-space neighborhoods. WRGAT (Suresh et al., 2021) learns edge re-weightings to expose distant similar nodes. GPR-GNN (Chien et al., 2020) and H2GCN (Zhu et al., 2020a) redesign the propagation operator to weight multi-hop signals adaptively.

Spectral methods. FAGCN (Bo et al., 2021) and ACM-GCN (Luan et al., 2021) retain high-frequency components that standard GCNs filter out, preserving the feature differences between heterophilic neighbors.

Global structure modeling. LINKX (Lim et al., 2021) and GloGNN (Li et al., 2022) drop the local propagation prior entirely and learn from the full adjacency or all-pairs feature similarity, recognizing that for sufficiently heterophilic graphs the local prior is harmful rather than merely uninformative.

Mixture-of-experts and decoupled architectures. A recent line of work uses gated mixtures or decoupled feature/structure pathways to handle nodes with different homophily levels within the same graph. These methods motivate our two-branch design but operate within a single representation space and combine signals at the same granularity.

Position of SimPlex-GT. Existing heterophily methods entangle local and global signals in a shared representation space, on the implicit assumption that the two can be jointly optimized at the same level of abstraction. We argue—and verify experimentally on graphs with mixed homophily (Penn94, Amazon-Ratings)—that this entanglement is the source of the unstable cross-regime performance these methods exhibit. SimPlex-GT instead decouples local and global reasoning by operating on them at different abstraction levels: pointwise message passing at the node level, attention at the prototype level. Conceptually, this is a discriminative analogue of latent-group models such as stochastic block models, where prototype assignments play the role of latent communities but are learned end-to-end against a downstream task.

C.3 Self-Supervised Graph Representation Learning

Self-supervised learning (SSL) on graphs has matured along two main axes: contrastive and reconstructive.

Contrastive. DGI (Veličković et al., 2019), MVGRL (Hassani & Khasahmadi, 2020), and GRACE (Zhu et al., 2020b) maximize agreement between augmented graph views, typically using InfoNCE-style objectives with negative sampling. Heterophily-aware variants such as HGRL (Chen et al., 2022a), GREET (Liu et al., 2022), MUSE (Yuan et al., 2023), and S3GCL (Wan et al., 2024) introduce topology-aware augmentations or explicit local/global view separation. These methods are sensitive to augmentation design and negative-sample selection.

Reconstructive. GAE/VGAE (Kipf & Welling, 2016b) reconstruct adjacency; GraphMAE (Hou et al., 2022) reconstructs masked features in the spirit of MAE (He et al., 2022). DSSL (Xiao et al., 2022) introduces structural decoupling for heterophilic graphs. Reconstruction objectives are tractable but not always aligned with downstream classification accuracy.

Negative-free latent prediction. A third line, originating with BYOL (Grill et al., 2020) and DINO (Caron et al., 2021) in vision, dispenses with negatives entirely by predicting the output of a slowly-evolving (EMA) teacher from a student view. BGRL (Thakoor et al., 2021) adapts BYOL to graphs.

Table 8: Node Clustering results (ACC in percent ± standard deviation). The best, runner-up and third best results for each dataset are highlighted.

Category	Methods	Texas	Actor	Cornell	CiteSeer
	Homo Ratio	0.11	0.22	0.30	0.74
Trad.	AE (Hinton & Salakhutdinov, 2006)	50.49±0.01	24.19±0.11	52.19±0.01	58.79±0.19
	node2vec (Grover & Leskovec, 2016)	48.80±1.93	25.02±0.04	50.98±0.01	20.76±0.27
	struc2vec (Ribeiro et al., 2017)	49.73±0.01	22.49±0.34	32.68±0.01	21.22±0.45
	LINE (Tang et al., 2015)	49.40±2.08	22.70±0.08	34.10±0.77	28.42±0.88
Attr.	GAE (Kipf & Welling, 2016b)	42.02±1.22	23.45±0.04	43.72±1.25	48.37±0.37
	VGAE (Kipf & Welling, 2016b)	50.27±1.87	23.30±0.22	43.39±0.99	55.67±0.13
	GraphSAGE (Hamilton et al., 2017)	56.83±0.56	23.08±0.29	44.70±2.00	49.28±1.18
	SDCN (Bo et al., 2020)	44.04±0.56	23.67±0.29	36.94±2.00	59.86±1.18
Homo	MVGRL (Hassani & Khasahmadi, 2020)	62.79±2.33	28.58±1.03	43.77±3.03	45.67±9.08
	GRACE (Zhu et al., 2020b)	56.99±2.23	25.87±0.45	43.55±4.60	54.66±5.41
	BGRL (Thakoor et al., 2021)	58.68±1.80	28.20±0.27	55.08±1.68	64.27±1.68
Hetero	DSSL (Xiao et al., 2022)	57.43±3.51	26.15±0.46	44.70±2.44	54.32±3.69
	HGRL (Chen et al., 2022a)	61.97±3.10	29.79±1.11	60.56±3.72	61.14±1.49
	† MUSE (Yuan et al., 2023)	65.79±4.36	31.05±0.72	62.35±2.38	66.03±2.33
Ours	SimPlex-GT -CF	76.36±1.86	32.66±0.56	73.91±2.72	66.82±2.26
	SimPlex-GT -CS	76.40±2.12	33.20±0.98	74.68±2.78	66.67±1.98

† MUSE doesn't provide any hyperparameters for node clustering.

Position of SimPlex-GT. Our self-supervised component follows the BYOL/BGRL recipe of EMA teacher–student matching with a masked student view, and is not a contribution in its own right. Its role here is targeted: it is the label-free pretraining objective of SimPlex-GT, evaluated with a frozen linear probe as for the SSL baselines, and because it backpropagates through the assignments, it also stabilizes prototype formation. We isolate this effect empirically (Sec. 4.2, App. N): with SSL, the prototypes stay diverse and in use (pairwise similarity falls from about 0.98 to 0.31–0.37 and usage perplexity recovers toward M). The two design choices that go beyond standard BGRL—discrepancy-guided masking that focuses on hard nodes, and orthogonality regularization between local and global branches—are minor additions; the conceptual debt to BYOL/BGRL is explicit.

D Performance Comparison for Node Clustering

We provide additional performance comparison for node clustering and graph classification. We consider four groups of baseline methods:

- *Traditional Clustering Methods (Structure- or Attribute-only):* AE (Hinton & Salakhutdinov, 2006), node2vec (Grover & Leskovec, 2016), struc2vec (Ribeiro et al., 2017), and LINE (Tang et al., 2015).
- *Attributed Graph Clustering Methods (Both attributes and topology):* GAE (VGAE) (Kipf & Welling, 2016b), GraphSAGE (Hamilton et al., 2017), and SDCN (Bo et al., 2020).
- *Self-Supervised Methods for Homophilic Graphs:* MVGRL (Hassani & Khasahmadi, 2020), GRACE (Zhu et al., 2020b), and BGRL (Thakoor et al., 2021).
- *Self-Supervised Methods for Heterophilic Graphs:* DSSL (Xiao et al., 2022), HGRL (Chen et al., 2022a), and MUSE (Yuan et al., 2023).

For a fair comparison, we follow a unified evaluation protocol: the generated node embeddings are used for k-means clustering. For MUSE, since hyperparameters are not fully specified in the original paper, we tune them on the validation set. For others, we report the results provided in the baseline papers (Chen et al., 2022a; Yuan et al., 2023).

From the results in the table, we observe that our SimPlex-GT consistently achieves superior performance compared to all baselines. In particular, SimPlex-GT outperforms existing methods by a large margin on the Texas and Cornell datasets. Despite the presence of complex and mixed structural patterns, SimPlex-GT still surpasses MUSE on the Actor dataset by 2.15%, and attains comparable performance on CiteSeer. These observations are consistent with the trends observed in the node classification experiments, indicating that the embeddings learned by SimPlex-GT exhibit high representational quality and strong generalization ability across different downstream tasks.

E Performance Comparison for Graph Classification

Similarly, we also report results for graph classification. To ensure consistency with the other experiments, we consider two categories of baseline models: (1) Classical GNNs, including GCN and GIN, and (2) Graph Transformers, including GraphGPS and Exphormer.

We evaluate these methods on three representative datasets: Peptides-func from the Long-Range Graph Benchmark (LRGB) (Dwivedi et al., 2022), and ogbg-molhiv and ogbg-molpcba from OGB (Hu et al., 2020; 2021). The evaluation metrics corresponding to each dataset are reported in the table. We use global mean pooling as the readout function across all baselines to ensure a fair comparison (Luo et al., 2025).

From the table, we observe that our method consistently outperforms both classical GNNs and graph transformers across all datasets, demonstrating the effectiveness and superior generalization capability of the proposed model.

Table 9: Graph classification results. Higher is better. Metrics: AP for Peptides-func and ogbg-molpcba; ROC-AUC for ogbg-molhiv.

Methods	Peptides-func	ogbg-molhiv	ogbg-molpcba
Metric	AP	ROC-AUC	AP
GCN	68.60±0.50	76.06±0.97	20.20±0.24
GIN	66.21±0.67	78.35±1.25	22.66±0.28
GraphGPS	65.35±0.41	78.80±1.01	29.07±0.28
Exphormer	62.58±0.92	78.34±0.44	28.49±0.25
SimPlex-GT -CF	71.67±0.78	80.42±1.35	29.80±0.29
SimPlex-GT -CS	72.12±0.86	79.86±1.28	29.22±0.26

F Proof of Theorem F.1

F.1 Definition and Setup

Consider a node classification task on graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node features $\mathbf{X} \in \mathbb{R}^{n \times d}$ and labels $\mathbf{y} \in \{-1, +1\}^n$.

Definition 1 (Node-level Homophily)**.** *For node $i \in \mathcal{V}$, define its local homophily ratio as:*

$$\rho_i = \frac{|\{j \in \mathcal{N}(i) : y_j = y_i\}|}{|\mathcal{N}(i)|} \tag{11}$$

Node i is locally heterophilic if $\rho_i < 0.5$.

Definition 2 (Heterophilic Node Set)**.** *Define the set of heterophilic nodes as:*

$$\mathcal{V}_{\text{hetero}} = \{i \in \mathcal{V} : \rho_i < 0.5\} \tag{12}$$

F.2 Main Theorem

To isolate the core message-passing mechanism from feature transformation effects, we first introduce several assumptions to enable a cleaner comparison between message-passing GNNs and graph transformers.

Theorem F.1 (GT vs. message passing under heterophily)**.** *Assume features align with labels in graphs: $x^{(0)}(i) = y_i \mathbf{e}$ with $\|\mathbf{e}\| = 1$. Let node-wise $\rho_i := \frac{|\{j \in \mathcal{N}(i):\, y_j = y_i\}|}{|\mathcal{N}(i)|}$, $\mathcal{V}_{\text{hetero}} = \{i : \rho_i < \frac{1}{2}, i \in \mathcal{V}\}$, $\mathbf{h}^{(1)}(i)$ is the node feature computed by a model (GNN or GT).*

- ***MP.*** *For a one-layer message passing with mean-aggregation without self-loop, $\langle \mathbf{h}^{(1)}(i), y_i \mathbf{e} \rangle = 2\rho_i - 1 < 0$ for all $i \in \mathcal{V}_{\text{hetero}}$, so it misclassifies at least $|\mathcal{V}_{\text{hetero}}|/|\mathcal{V}|$ nodes.*
- ***GT.*** *There exist parameters $\boldsymbol{\theta}^*$ such that: $\langle \mathbf{h}^{(1)}_{\boldsymbol{\theta}^*}(i), y_i \mathbf{e} \rangle > 0, \forall i \in \mathcal{V}$, yielding zero error.*

Proof. **Message-Passing (MP):**

Consider one-layer message passing with mean aggregation at node i (without self-loop):

$$\mathbf{h}^{(1)}(i) = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathbf{x}^{(0)}(j). \tag{13}$$

Under the assumption $\mathbf{x}^{(0)}(j) = y_j \mathbf{e}$, we have

$$\mathbf{h}^{(1)}(i) = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} y_j \mathbf{e}. \tag{14}$$

Decomposing the neighborhood into same-class and different-class neighbors:

$$\mathbf{h}_i^{(1)} = \frac{1}{|\mathcal{N}(i)|} \left(\sum_{j \in \mathcal{N}(i): y_j = y_i} y_i \mathbf{e} + \sum_{j \in \mathcal{N}(i): y_j \neq y_i} (-y_i) \mathbf{e} \right) \tag{15}$$

$$= \frac{1}{|\mathcal{N}(i)|} \left(\rho_i |\mathcal{N}(i)| \cdot y_i \mathbf{e} + (1 - \rho_i) |\mathcal{N}(i)| \cdot (-y_i) \mathbf{e} \right) \tag{16}$$

$$= y_i (2\rho_i - 1) \mathbf{e} \tag{17}$$

Therefore,

$$\langle \mathbf{h}^{(1)}(i), y_i \mathbf{e} \rangle = (2\rho_i - 1) \|\mathbf{e}\|^2 = 2\rho_i - 1. \tag{18}$$

For any $i \in \mathcal{V}_{\text{hetero}}$ where $\rho_i < \frac{1}{2}$, we obtain $\langle \mathbf{h}^{(1)}(i), y_i \mathbf{e} \rangle < 0$. Hence, using the linear decoding rule $\hat{y}_i = \operatorname{sign}\big(\langle \mathbf{h}^{(1)}(i), \mathbf{e} \rangle\big)$, MP misclassifies all nodes in $\mathcal{V}_{\text{hetero}}$, leading to error at least $|\mathcal{V}_{\text{hetero}}|/|\mathcal{V}|$.

Graph Transformer: Set

$$\mathbf{W}_Q = \mathbf{W}_K = \gamma \, \mathbf{I}, \qquad \mathbf{W}_V = \mathbf{I}, \quad \gamma > 0. \tag{19}$$

Then

$$\alpha_{ij} \propto \exp\left(\frac{(\mathbf{W}_Q \mathbf{x}_i^{(0)})^T (\mathbf{W}_K \mathbf{x}_j^{(0)})}{\sqrt{d}} \right) = \exp\left(\frac{\gamma^2 \, (\mathbf{x}_i^{(0)})^T \mathbf{x}_j^{(0)}}{\sqrt{d}} \right) = \exp\left(\frac{\gamma^2 \, y_i y_j \, \|\mathbf{e}\|^2}{\sqrt{d}} \right). \tag{20}$$

Hence

$$\alpha_{ij} \propto \begin{cases} \exp\left(\dfrac{\gamma^2 \|\mathbf{e}\|^2}{\sqrt{d}} \right) & \text{if } y_i = y_j, \\ \exp\left(-\dfrac{\gamma^2 \|\mathbf{e}\|^2}{\sqrt{d}} \right) & \text{if } y_i \neq y_j. \end{cases} \tag{21}$$

Sharpening limit via γ. Split the normalizer into same/different-label parts and divide through by $\exp(\gamma^2 \|\mathbf{e}\|^2 / \sqrt{d})$:

$$\alpha_{ij} = \begin{cases} \dfrac{1}{|\{k : y_k = y_i\}| + |\{k : y_k \neq y_i\}| \, \exp\big(-\frac{2\gamma^2 \|\mathbf{e}\|^2}{\sqrt{d}} \big)} & \text{if } y_j = y_i, \\ \dfrac{\exp\big(-\frac{2\gamma^2 \|\mathbf{e}\|^2}{\sqrt{d}} \big)}{|\{k : y_k = y_i\}| + |\{k : y_k \neq y_i\}| \, \exp\big(-\frac{2\gamma^2 \|\mathbf{e}\|^2}{\sqrt{d}} \big)} & \text{if } y_j \neq y_i. \end{cases} \tag{22}$$

As $\gamma \to \infty$ (effective sharpening increases), we have $\exp\big(-\frac{2\gamma^2 \|\mathbf{e}\|^2}{\sqrt{d}} \big) \to 0$, and therefore

$$\alpha_{ij} \to \begin{cases} \dfrac{1}{|\{k : y_k = y_i\}|} & \text{if } y_i = y_j, \\ 0 & \text{if } y_i \neq y_j. \end{cases} \tag{23}$$

Consequently,

$$\mathbf{h}_i^{(1)} = \sum_{j \in \mathcal{V}} \alpha_{ij} \, \mathbf{x}_j^{(0)} = \sum_{j : y_j = y_i} \alpha_{ij} \, y_j \mathbf{e} = y_i \mathbf{e} \sum_{j : y_j = y_i} \alpha_{ij} = y_i \mathbf{e}, \tag{24}$$

and thus

$$\langle \mathbf{h}_i^{(1)}, y_i \mathbf{e} \rangle = \|\mathbf{e}\|^2 > 0. \tag{25}$$

We can conclude that even under an idealized setting, classical GNNs will still fail due to the structural noise introduced by heterophilic edges, whereas a Graph Transformer will not. The key reason for the degraded performance on heterophilic graphs lies in the **message-passing mechanism** itself.

□

F.3 Performance Bounds

Corollary. *Define the heterophilic node set $\mathcal{V}_{\text{hetero}} = \{\, i : \rho_i < \frac{1}{2} \,\}$. Under the conditions of Theorem F.1:*

1. ***GNN:** A one-layer mean-aggregation (linearized) GNN misclassifies every node with $\rho_i < \frac{1}{2}$; hence it misclassifies at least a $|\mathcal{V}_{\text{hetero}}|/|\mathcal{V}|$ fraction of nodes.*
2. ***Graph Transformer:** With the attention in Theorem F.1 (i.e., the sharpening limit), the first-layer outputs satisfy $\langle \mathbf{h}_i^{(1)}, y_i \mathbf{e} \rangle > 0$ for all i, yielding zero classification error.*

G Proof of Theorem 1

Theorem G.1 (Variance reduction via soft clusters)**.** *Assume $x_i^{(0)} = y_i \mathbf{e} + \varepsilon_i$ with $\mathbb{E}[\varepsilon_i] = 0$ and $\mathrm{Cov}(\varepsilon_i) = \Sigma \preceq \sigma^2 I$. Let the soft assignment matrix be $p_{i,c} \in (0,1)$ with $\sum_{c=1}^{M} p_{i,c} = 1$ for each node i. Define the (soft) cluster mass and prototype by*

$$s_c \triangleq \sum_{i=1}^{N} p_{i,c}, \tag{26}$$

$$\mathbf{p}_c \triangleq \frac{1}{s_c} \sum_{i=1}^{N} p_{i,c}\, x_i^{(0)}. \tag{27}$$

Let the Node-to-Cluster (N2C) output be $h_i^{\mathrm{N2C}} \triangleq \sum_{c=1}^{M} \alpha_{i\to c}\, \mathbf{p}_c$ where $\alpha_{i\to c} \geq 0$ and $\sum_{c=1}^{M} \alpha_{i\to c} = 1$. Then, along direction $\mathbf{e}$, the following bounds hold (without assuming independence across clusters):

$$\mathrm{Var}\big(\langle \mathbf{p}_c, \mathbf{e} \rangle\big) \;\leq\; \sigma^2 \|\mathbf{e}\|^2 \cdot \frac{\sum_{i=1}^{N} p_{i,c}^2}{\Big(\sum_{i=1}^{N} p_{i,c}\Big)^2}, \tag{28}$$

$$\mathrm{Var}\big(\langle h_i^{\mathrm{N2C}}, \mathbf{e} \rangle\big) \;\leq\; \sum_{c=1}^{M} \alpha_{i\to c}\, \sigma^2 \|\mathbf{e}\|^2 \cdot \frac{\sum_{i=1}^{N} p_{i,c}^2}{\Big(\sum_{i=1}^{N} p_{i,c}\Big)^2}. \tag{29}$$

By contrast, a single-neighbor Node-to-Node readout satisfies

$$\mathrm{Var}\big(\langle x_j^{(0)}, \mathbf{e} \rangle\big) = \mathrm{Var}\big(\langle \varepsilon_j, \mathbf{e} \rangle\big) = \mathbf{e}^\top \Sigma \mathbf{e} \;\leq\; \sigma^2 \|\mathbf{e}\|^2. \tag{30}$$

Proof. **Step 1: Reweighting form of the prototype.** Let $w_{i,c} \triangleq p_{i,c}/s_c$ so that $\sum_{i=1}^{N} w_{i,c} = 1$. Then

$$\mathbf{p}_c \;=\; \sum_{i=1}^{N} w_{i,c}\, x_i^{(0)} \;=\; \sum_{i=1}^{N} w_{i,c}\, (y_i \mathbf{e} + \varepsilon_i), \tag{31}$$

$$\langle \mathbf{p}_c, \mathbf{e} \rangle \;=\; \sum_{i=1}^{N} w_{i,c}\, \big(y_i \|\mathbf{e}\|^2 + \langle \varepsilon_i, \mathbf{e} \rangle\big). \tag{32}$$

The class-mean term $\sum_i w_{i,c} y_i \|\mathbf{e}\|^2$ is deterministic w.r.t. noise, hence

$$\mathrm{Var}\big(\langle \mathbf{p}_c, \mathbf{e} \rangle\big) \;=\; \mathrm{Var}\left(\sum_{i=1}^{N} w_{i,c}\, \langle \varepsilon_i, \mathbf{e} \rangle \right). \tag{33}$$

Step 2: Bounding the prototype variance. Assuming independence across nodes and $\mathrm{Cov}(\varepsilon_i) = \Sigma \preceq \sigma^2 I$,

$$\mathrm{Var}\left(\sum_{i=1}^{N} w_{i,c}\, \langle \varepsilon_i, \mathbf{e} \rangle \right) \;=\; \sum_{i=1}^{N} w_{i,c}^2\, \mathrm{Var}\big(\langle \varepsilon_i, \mathbf{e} \rangle\big) \;=\; \big(\mathbf{e}^\top \Sigma \mathbf{e}\big) \sum_{i=1}^{N} w_{i,c}^2, \tag{34}$$

$$\mathrm{Var}\big(\langle \mathbf{p}_c, \mathbf{e} \rangle\big) \;\leq\; \sigma^2 \|\mathbf{e}\|^2 \sum_{i=1}^{N} w_{i,c}^2 \;=\; \sigma^2 \|\mathbf{e}\|^2 \cdot \frac{\sum_{i=1}^{N} p_{i,c}^2}{\Big(\sum_{i=1}^{N} p_{i,c}\Big)^2}. \tag{35}$$

Step 3: N2C output as a convex combination (correlated clusters). Write $X_c \triangleq \langle \mathbf{p}_c, \mathbf{e} \rangle$. Then

$$\langle h_i^{\mathrm{N2C}}, \mathbf{e} \rangle \;=\; \sum_{c=1}^{M} \alpha_{i\to c}\, X_c, \qquad \alpha_{i\to c} \ge 0,\;\; \sum_{c=1}^{M} \alpha_{i\to c} = 1. \tag{36}$$

We bound the variance in two steps, neither of which assumes independence across clusters. First, by Cauchy–Schwarz on covariances ($|\mathrm{Cov}(X_c, X_d)| \le \sqrt{\mathrm{Var}(X_c)\,\mathrm{Var}(X_d)}$),

$$\mathrm{Var}\Big(\sum_{c=1}^{M} \alpha_{i\to c}\, X_c\Big) \;=\; \sum_{c,d=1}^{M} \alpha_{i\to c}\,\alpha_{i\to d}\,\mathrm{Cov}(X_c, X_d) \;\le\; \Big(\sum_{c=1}^{M} \alpha_{i\to c}\,\sqrt{\mathrm{Var}(X_c)}\Big)^2. \tag{37}$$

Second, by Jensen's inequality applied to the convex function $\phi(x) = x^2$ with weights $\alpha_{i\to c} \ge 0$ summing to 1,

$$\Big(\sum_{c=1}^{M} \alpha_{i\to c}\,\sqrt{\mathrm{Var}(X_c)}\Big)^2 \;\le\; \sum_{c=1}^{M} \alpha_{i\to c}\,\mathrm{Var}(X_c). \tag{38}$$

Substituting the Step 2 bound $\mathrm{Var}(X_c) \le \sigma^2\|\mathbf{e}\|^2 \cdot \sum_i p_{i,c}^2/(\sum_i p_{i,c})^2$ yields

$$\mathrm{Var}\big(\langle h_i^{\mathrm{N2C}}, \mathbf{e}\rangle\big) \;\le\; \sum_{c=1}^{M} \alpha_{i\to c}\,\sigma^2\|\mathbf{e}\|^2 \cdot \frac{\sum_{i=1}^{N} p_{i,c}^2}{\Big(\sum_{i=1}^{N} p_{i,c}\Big)^2}. \tag{39}$$

Step 4: N2N (single neighbor) comparison. For any single-neighbor readout,

$$\mathrm{Var}\big(\langle x_j^{(0)}, \mathbf{e}\rangle\big) \;=\; \mathrm{Var}\big(\langle \varepsilon_j, \mathbf{e}\rangle\big) \;=\; \mathbf{e}^\top \Sigma \mathbf{e} \;\le\; \sigma^2\|\mathbf{e}\|^2. \tag{40}$$

Let

$$R \;\triangleq\; \sum_{c=1}^{M} \alpha_{i\to c}\,\sigma^2\|\mathbf{e}\|^2\, \frac{\sum_{j=1}^{N} p_{j,c}^2}{\Big(\sum_{j=1}^{N} p_{j,c}\Big)^2} \;=\; \sigma^2\|\mathbf{e}\|^2 \sum_{c=1}^{M} \alpha_{i\to c}\, r_c, \qquad r_c \;\triangleq\; \frac{\sum_j p_{j,c}^2}{s_c^2},\;\; s_c \triangleq \sum_j p_{j,c}. \tag{41}$$

Then R is the variance upper bound from Step 3. We now show why $R \le \sigma^2\|\mathbf{e}\|^2$.

Bounds on r_c. Define normalized weights $w_{j,c} = \frac{p_{j,c}}{s_c}$. Then:

- $w_{j,c} \ge 0$ and $\sum_j w_{j,c} = 1$ (forms a probability distribution)
- $0 \le w_{j,c} \le 1$ for all j (since $p_{j,c} \le s_c$ always holds)

Therefore:

$$r_c = \frac{\sum_j p_{j,c}^2}{s_c^2} = \sum_j \left(\frac{p_{j,c}}{s_c}\right)^2 = \sum_j w_{j,c}^2 \tag{42}$$

Since $0 \le w_{j,c} \le 1$, we have $w_{j,c}^2 \le w_{j,c}$, thus:

$$r_c = \sum_j w_{j,c}^2 \le \sum_j w_{j,c} = 1 \tag{43}$$

Convex combination. Since $\alpha_{i\to c} \ge 0$ and $\sum_c \alpha_{i\to c} = 1$ (attention weights form a probability distribution):

$$\sum_{c=1}^{M} \alpha_{i\to c} \cdot r_c \le \sum_{c=1}^{M} \alpha_{i\to c} \cdot 1 = 1 \tag{44}$$

Hence:

$$R = \sigma^2\|\mathbf{e}\|^2 \sum_c \alpha_{i\to c} r_c \le \sigma^2\|\mathbf{e}\|^2 \tag{45}$$

The inequality $r_c < 1$ is strict whenever at least two nodes contribute to cluster c □

Corollary G.1 (Hierarchical logit stabilization)**.** *Let the attention logit be an inner product with the query q_i. Define the unit direction $e := q_i/\|q_i\|$. Then the variance bounds along q_i for any cluster*

prototype p_c,

$$\mathrm{Var}\big(\langle p_c, q_i\rangle\big) \;\le\; \sigma^2\|q_i\|^2\, r_c,$$

where $r_c := \frac{\sum_j p_{j,c}^2}{\left(\sum_j p_{j,c}\right)^2} \le 1$. By contrast, for a node-level (N2N) logit using a single node $x_j^{(0)}$,

$$\mathrm{Var}\big(\langle x_j^{(0)}, q_i\rangle\big) \;\le\; \sigma^2\|q_i\|^2.$$

Consequently, N2C reduces logit variance by a multiplicative factor r_c, making the subsequent softmax weighting less sensitive to node-level noise.

Proof. From the identity $\langle x, q_i\rangle = \|q_i\|\langle x, e\rangle$ with $e = q_i/\|q_i\|$, we have $\mathrm{Var}(\langle x, q_i\rangle) = \|q_i\|^2\,\mathrm{Var}(\langle x, e\rangle)$. Applying Theorem 2's bounds $\mathrm{Var}(\langle p_c, e\rangle) \le \sigma^2\|e\|^2 r_c$ and $\mathrm{Var}(\langle x_j^{(0)}, e\rangle) \le \sigma^2\|e\|^2$ (with $\|e\| = 1$) yields the stated inequalities. □

Effective cluster size. Let $s_c := \sum_i p_{i,c}$ and $w_{i,c} := p_{i,c}/s_c$, so that $\sum_i w_{i,c} = 1$ and $0 \le w_{i,c} \le 1$. Define the effective cluster size $n_{\mathrm{eff},c} := 1/\sum_i w_{i,c}^2$. Then

$$r_c \;=\; \frac{\sum_i p_{i,c}^2}{\left(\sum_i p_{i,c}\right)^2} \;=\; \sum_i w_{i,c}^2 \;=\; 1/n_{\mathrm{eff},c}, \tag{46}$$

and the bounds of Theorem G.1 can be rewritten as

$$\mathrm{Var}\big(\langle \mathbf{p}_c, \mathbf{e}\rangle\big) \;\le\; \frac{\sigma^2\|\mathbf{e}\|^2}{n_{\mathrm{eff},c}}, \qquad \mathrm{Var}\Big(\sum_c \alpha_{i\to c}\,\langle \mathbf{p}_c, \mathbf{e}\rangle\Big) \;\le\; \sigma^2\|\mathbf{e}\|^2 \sum_c \frac{\alpha_{i\to c}}{n_{\mathrm{eff},c}}. \tag{47}$$

N2C therefore provides denoising proportional to the effective cluster size, whereas N2N has no such $1/n_{\mathrm{eff}}$ factor.

H Proof of Theorem H.1

Lemma 1 (Near-orthogonality in ℓ_2)**.** *Let $L = U\Lambda U^\top$ with $U^\top U = I$, and define $G = U\,g(\Lambda)\,U^\top$, $H = I - G = U\,(I - g(\Lambda))\,U^\top$ for a response $g : [0,\infty) \to [0,1]$. For $x \in \mathbb{R}^{N\times d}$, write $\hat{X} := U^\top x$ (its k-th row is $\hat{x}_k^\top$). Then*

$$\langle x_g, x_f\rangle := \mathrm{Tr}(x_g^\top x_f) = \sum_{k=1}^{N} g(\lambda_k)\,\big(1 - g(\lambda_k)\big)\,\|\hat{x}_k\|_2^2 \;\le\; \varepsilon\,\|x\|_F^2, \tag{48}$$

where $\varepsilon := \max_{\lambda\ge 0} g(\lambda)(1-g(\lambda)) \le \frac{1}{4}$, and equality $\langle x_g, x_f\rangle = 0$ holds if $g(\lambda) \in \{0,1\}$ for all λ (spectral projector).

Proof. By definition $x_g = Gx = U\,g(\Lambda)\,\hat{X}$ and $x_f = Hx = U\,(I - g(\Lambda))\,\hat{X}$. Using $U^\top U = I$ and cyclicity of trace,

$$\begin{aligned}\langle x_g, x_f\rangle &= \mathrm{Tr}\Big((Ug(\Lambda)\hat{X})^\top (U(I-g(\Lambda))\hat{X})\Big) = \mathrm{Tr}\Big(\hat{X}^\top g(\Lambda)(I-g(\Lambda))\hat{X}\Big)\\ &= \sum_{k=1}^{N} g(\lambda_k)\big(1-g(\lambda_k)\big)\,\|\hat{x}_k\|_2^2,\end{aligned} \tag{49}$$

since $g(\Lambda)$ and $I - g(\Lambda)$ are diagonal and multiply elementwise in the eigenbasis. Because $0 \le g(\lambda) \le 1$, we have $0 \le g(\lambda)(1-g(\lambda)) \le \varepsilon \le \frac{1}{4}$, hence

$$|\langle x_g, x_f\rangle| \le \varepsilon \sum_{k=1}^{N} \|\hat{x}_k\|_2^2 = \varepsilon\,\|x\|_F^2, \tag{50}$$

using $\|x\|_F^2 = \|\hat{X}\|_F^2$ (orthonormal U preserves the Frobenius norm). If g is a spectral projector, each factor $g(\lambda_k)(1-g(\lambda_k)) = 0$, so $\langle x_g, x_f\rangle = 0$. □

Lemma 2 (Variance reduction via contraction)**.** *Let $\eta \in \mathbb{R}^N$ be a zero-mean random vector with covariance $\mathrm{Cov}(\eta) \preceq \sigma^2 I$. For any unit $\mathbf{e} \in \mathbb{R}^N$,*

$$\mathrm{Var}\big(\langle G\eta, \mathbf{e}\rangle\big) \;=\; \mathbf{e}^\top G\,\mathrm{Cov}(\eta)\,G\,\mathbf{e} \;\le\; \sigma^2\,\mathbf{e}^\top G^2\mathbf{e} \;\le\; \sigma^2\,\mathbf{e}^\top G\mathbf{e} \;\le\; \sigma^2. \tag{51}$$

The inequalities are strict whenever $g(\lambda) \in (0,1)$ on a set of eigenvalues with nonzero variance mass along v. An analogous statement holds for $H = I - G$.

Proof. First, $\langle G\eta, \mathbf{e}\rangle = \langle \eta, G^\top \mathbf{e}\rangle$ and G is symmetric, so

$$\mathrm{Var}(\langle G\eta, \mathbf{e}\rangle) = \mathrm{Var}(\langle \eta, G\mathbf{e}\rangle) = (G\mathbf{e})^\top \mathrm{Cov}(\eta)\,(G\mathbf{e}) = \mathbf{e}^\top G\,\mathrm{Cov}(\eta)\,G\,\mathbf{e}. \tag{52}$$

Using $\mathrm{Cov}(\eta) \preceq \sigma^2 I$ gives $v^\top G\,\mathrm{Cov}(\eta)\,G\,v \le \sigma^2\, v^\top G^2 v$.

Now write $G = U\,g(\Lambda)\,U^\top$. Then

$$\mathbf{e}^\top G^2 \mathbf{e} = \mathbf{e}^\top U g(\Lambda)^2 U^\top \mathbf{e} = \sum_{k=1}^{N} g(\lambda_k)^2\,\xi_k^2, \qquad \mathbf{e}^\top G \mathbf{e} = \sum_{k=1}^{N} g(\lambda_k)\,\xi_k^2, \tag{53}$$

where $\xi = U^\top \mathbf{e}$ and $\sum_k \xi_k^2 = \|\mathbf{e}\|^2 = 1$. Since $0 \le g(\lambda) \le 1$, we have $g(\lambda)^2 \le g(\lambda)$, hence $\mathbf{e}^\top G^2 \mathbf{e} \le \mathbf{e}^\top G \mathbf{e} \le \sum_k \xi_k^2 = 1$. Combining the steps yields the chain of inequalities.

Strictness: if there exists k with $\xi_k \neq 0$, $g(\lambda_k) \in (0,1)$ and positive variance along that eigen-direction (i.e., the bound $\mathrm{Cov}(\eta) \preceq \sigma^2 I$ is not tight as zero there), then at least one inequality becomes strict. The same argument applies to $H = U(I - g(\Lambda))U^\top$ since its eigenvalues lie in $[0,1]$ and are $1 - g(\lambda)$. □

Theorem H.1 (Stability in CF)**.** *Let keys/values be computed from low-pass prototypes and queries from high-pass features:*

$$s_{i,c} := \frac{1}{\sqrt{d_k}}\,\big\langle Q(x_f^{(i)}),\, K(c_b^{(c)})\big\rangle, \qquad s^\star_{i,c} := \frac{1}{\sqrt{d_k}}\,\big\langle Q(h^{(i)}),\, K(s^{(c)})\big\rangle, \tag{54}$$

where $h^{(i)}$ is the high-frequency component of node i, and $s^{(c)}$ the low-frequency prototype of cluster c. Assume Q and K are Lipschitz: $\|Q(a) - Q(b)\| \le L_Q\|a-b\|$, $\|K(a) - K(b)\| \le L_K\|a-b\|$, and that $\|K(c_b^{(c)})\| \le M_K$ almost surely (e.g., keys followed by LayerNorm or a bounded activation), $\mathbb{E}\|Q(h^{(i)})\|^2 \le M_Q^2$. Then

$$\mathbb{E}\big[(s_{i,c} - s^\star_{i,c})^2\big] \;\le\; \frac{2}{d_k}\Big(L_Q^2 M_K^2\,\mathbb{E}\|x_f^{(i)} - h^{(i)}\|^2 \;+\; L_K^2 M_Q^2\,\mathbb{E}\|c_b^{(c)} - s^{(c)}\|^2\Big). \tag{55}$$

Consequently, by Lemma 2, both error terms on the right shrink under the contractions H and G, yielding strictly smaller logit MSE than using unsplit features whenever the filters attenuate off-target spectra.

When the contraction holds. The strict-improvement part of Theorem H.1 is not automatic. $\hat{A}$ has eigenvalues in $(-1, 1]$, so the high-pass operator $I - \hat{A}$ has eigenvalues in $[0, 2)$: it contracts on the non-negative spectrum of $\hat{A}$ but amplifies components along negative eigenvalues, which are more prominent on heterophilic, near-bipartite structure. The improvement therefore applies to the part of the noise lying in the non-negative spectrum of $\hat{A}$. The key-norm bound is required almost surely rather than in expectation because the query and key deviations are functions of the same noisy features.

Proof. Let $a := Q(x_f^{(i)})$, $a^\star := Q(h^{(i)})$, $b := K(c_b^{(c)})$, $b^\star := K(s^{(c)})$. Then

$$s_{i,c} - s^\star_{i,c} = \frac{1}{\sqrt{d_k}}\big(\langle a, b\rangle - \langle a^\star, b^\star\rangle\big) \tag{56}$$

$$= \frac{1}{\sqrt{d_k}}\Big(\underbrace{\langle a, b\rangle - \langle a^\star, b\rangle}_{=\,\langle a - a^\star,\, b\rangle} + \underbrace{\langle a^\star, b\rangle - \langle a^\star, b^\star\rangle}_{=\,\langle a^\star,\, b - b^\star\rangle}\Big) \tag{57}$$

$$= \frac{1}{\sqrt{d_k}}\Big(\langle a - a^\star,\, b\rangle + \langle a^\star,\, b - b^\star\rangle\Big). \tag{58}$$

By Cauchy–Schwarz,

$$|s_{i,c} - s^\star_{i,c}| \le \frac{1}{\sqrt{d_k}}\big(\|a - a^\star\|\,\|b\| + \|a^\star\|\,\|b - b^\star\|\big). \tag{59}$$

Square and use $(u+v)^2 \le 2(u^2 + v^2)$:

$$(s_{i,c} - s^\star_{i,c})^2 \le \frac{2}{d_k}\big(\|a - a^\star\|^2\,\|b\|^2 + \|a^\star\|^2\,\|b - b^\star\|^2\big). \tag{60}$$

Apply Lipschitz bounds $\|a - a^\star\| \le L_Q\|x_f^{(i)} - h^{(i)}\|$ and $\|b - b^\star\| \le L_K\|c_b^{(c)} - s^{(c)}\|$, then take expectation and use $\|b\|^2 \le M_K^2$ almost surely (so $\mathbb{E}[\|a - a^\star\|^2\|b\|^2] \le M_K^2\,\mathbb{E}\|a - a^\star\|^2$ without assuming independence of the two factors, which are both functions of the same noisy features) and

$\|a^\star\|^2 \le M_Q^2$, where $a^\star = Q(h^{(i)})$ is deterministic because $h^{(i)}$ is the noise-free target:

$$\mathbb{E}(s_{i,c} - s^\star_{i,c})^2 \le \frac{2}{d_k}\Big(L_Q^2\,\mathbb{E}\|x_f^{(i)} - h^{(i)}\|^2\,M_K^2 + L_K^2\,\mathbb{E}\|c_b^{(c)} - s^{(c)}\|^2\,M_Q^2\Big), \tag{61}$$

which yields the stated bound. Finally, Lemma 2 (applied to H on x and to G on prototypes) implies that both $\mathbb{E}\|x_f^{(i)} - h^{(i)}\|^2$ and $\mathbb{E}\|c_b^{(c)} - s^{(c)}\|^2$ strictly decrease whenever the corresponding filters have eigenvalues strictly inside $(0,1)$ on nontrivial spectral mass, so the right-hand side is strictly smaller than without complementary filtering. □

I Proof of Theorem I.1

Theorem I.1 (Variance reduction in CS)**.** *Let M be the number of soft clusters. For each cluster c, where $c \notin \mathcal{N}(c)$, unsmoothed prototype $z_c \in \mathbb{R}^d$ is written as $z_c = \mu_c + \varepsilon_c$, where μ_c is the cluster mean and ε_c is a zero-mean random vector with $\mathbb{E}[\varepsilon_c] = 0$ and $\mathrm{Cov}(\varepsilon_c) \preceq \sigma^2 I$. Let $\mathcal{N}(c)$ denote neighbors of c on the cluster graph G_c (e.g., $A_c = C^\top A C - \mathrm{diag}(C^\top A C)$ as in Eq. 5). Let weights $w_{cu} \ge 0$ satisfy $\sum_{u\in\mathcal{N}(c)} w_{cu} = 1$. We consider one-step residual low-pass smoothing: $\tilde z_c = (1-\alpha)\,z_c + \alpha \sum_{u\in\mathcal{N}(c)} w_{cu}\,z_u$, where $\alpha \in (0,1)$, and $\mathbf{e} \in \mathbb{R}^d$ denotes any unit vector, i.e., $\|\mathbf{e}\| = 1$. Assume that $\{\varepsilon_u\}_{u=1}^M$ are independent across clusters (exact for hard, non-overlapping assignments; see the remark after the proof for soft assignments). Then, for any unit direction $\mathbf{e}$,*

$$\mathrm{Var}\big(\langle \tilde z_c - \mu_c, \mathbf{e}\rangle\big) \;=\; \mathrm{Var}\Big(\big\langle (1-\alpha)\varepsilon_c + \alpha\sum\nolimits_u w_{cu}\varepsilon_u,\ \mathbf{e}\big\rangle\Big) \;\le\; \big((1-\alpha)^2 + \alpha^2\big)\,\sigma^2 \le\; \sigma^2, \tag{62}$$

with strict inequality whenever $0 < \alpha < 1$.

Proof. By the residual low-pass update and the noise model $z_c = \mu_c + \varepsilon_c$, $z_u = \mu_u + \varepsilon_u$,

$$\begin{aligned}\tilde z_c - \mu_c &= (1-\alpha)(\mu_c + \varepsilon_c) + \alpha\sum_u w_{cu}(\mu_u + \varepsilon_u) - \mu_c \\ &= \underbrace{\alpha\sum_u w_{cu}(\mu_u - \mu_c)}_{=:\ b_c\ \text{(structure bias)}} \;+\; \underbrace{(1-\alpha)\varepsilon_c + \alpha\sum_u w_{cu}\varepsilon_u}_{\text{noise part}}.\end{aligned} \tag{63}$$

Projecting onto an arbitrary unit direction $\mathbf{e} \in \mathbb{R}^d$ turns vectors into scalars. Define

$$X := \langle \varepsilon_c, \mathbf{e}\rangle, \qquad Y := \big\langle \sum_u w_{cu}\varepsilon_u,\ \mathbf{e}\big\rangle, \qquad \beta := \langle b_c, \mathbf{e}\rangle = \alpha\sum_u w_{cu}\,\langle \mu_u - \mu_c, \mathbf{e}\rangle. \tag{64}$$

Then we have the exact decomposition

$$\langle \tilde z_c - \mu_c, \mathbf{e}\rangle = (1-\alpha)X + \alpha Y + \beta. \tag{65}$$

Since β is a constant (given the means), it does not affect variance:

$$\mathrm{Var}\big(\langle \tilde z_c - \mu_c, \mathbf{e}\rangle\big) = \mathrm{Var}\big((1-\alpha)X + \alpha Y\big). \tag{66}$$

Using the variance decomposition formula gives

$$\mathrm{Var}\big((1-\alpha)X + \alpha Y\big) = (1-\alpha)^2\mathrm{Var}(X) + \alpha^2\mathrm{Var}(Y) + 2\alpha(1-\alpha)\mathrm{Cov}(X,Y). \tag{67}$$

Cross term. Assume noises are independent across clusters; then X is independent of Y and $\mathrm{Cov}(X,Y) = 0$.

Bounding each variance. From $\mathrm{Cov}(\varepsilon_c) \preceq \sigma^2 I$ and $\|\mathbf{e}\| = 1$,

$$\mathrm{Var}(X) = \mathrm{Var}(\langle \varepsilon_c, \mathbf{e}\rangle) = \mathbf{e}^\top \mathrm{Cov}(\varepsilon_c)\mathbf{e} \le \sigma^2. \tag{68}$$

Independence across different u yields

$$\mathrm{Var}(Y) = \mathrm{Var}\Big(\sum_u w_{cu}\langle \varepsilon_u, \mathbf{e}\rangle\Big) = \sum_u w_{cu}^2\,\mathrm{Var}(\langle \varepsilon_u, \mathbf{e}\rangle) \le \sigma^2\sum_u w_{cu}^2 \le \sigma^2\sum_u w_{cu} = \sigma^2, \tag{69}$$

where we used $0 \le w_{cu} \le 1$ and $\sum_u w_{cu} = 1$.

Combine. Plugging the bounds into the decomposition (with $\mathrm{Cov}(X,Y) = 0$) gives

$$\mathrm{Var}\big(\langle \tilde z_c - \mu_c, \mathbf{e}\rangle\big) \le \big((1-\alpha)^2 + \alpha^2\big)\,\sigma^2. \tag{70}$$

Finally, if neighboring means are similar along $\mathbf{e}$, the last (bias) term is negligible, and $((1-\alpha)^2 + \alpha^2) = 1 - 2\alpha + 2\alpha^2 < 1$ for $\alpha \in (0,1)$, which gives the stated reduction. □

Remark (soft assignments and normalization). Under the soft assignments of Eq. 1, every prototype is a weighted mean of the same node noise, so with independent node noise $\mathrm{Cov}(\varepsilon_i) \preceq \sigma_0^2 I$ the cross-cluster covariance is $\mathrm{Cov}(\varepsilon_c, \varepsilon_u) = \sum_i \tilde{C}_{ic}\tilde{C}_{iu}\mathrm{Cov}(\varepsilon_i) \succeq 0$ with $\tilde{C} = CD_C^{-1}$, and $\mathrm{Cov}(X, Y)$ is generally positive. Without the independence assumption, Cauchy–Schwarz gives $\mathrm{Cov}(X, Y) \leq \sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}$ and hence only the non-expansion bound $\mathrm{Var}(\langle \tilde{z}_c - \mu_c, \mathbf{e}\rangle) \leq \big((1-\alpha)+\alpha\big)^2\sigma^2 = \sigma^2$; the strict reduction above holds to the extent that the overlap term $\sum_i \tilde{C}_{ic}\tilde{C}_{iu}$ is small, i.e., when assignments are close to hard. The proof also uses row-stochastic weights; Eq. 5 normalizes A_c symmetrically, whose rows sum to one only when the coarse graph is regular. Last, the mean-squared error around μ_c is $\mathrm{Var} + \|b_c\|^2$: smoothing across clusters with different means (as on heterophilic coarse graphs) trades variance for bias, and α sets that trade-off.

J Controlled Component Ablations

Each comparison below changes a single component of SimPlex-GT-CS and keeps the architecture, prototype count M, hidden dimension, data splits, and optimization protocol fixed.

Input-conditioned vs. freely learned prototypes. Table 10 replaces $Z = D_C^{-1}C^\top X$ with M free parameter vectors. Accuracy drops on all four graphs, spanning homophily ratios 0.21–0.81. Caching changes only the inference cost: during training, the soft-mean construction ties each prototype to a population of nodes, and CF, CS, and the assignment-level SSL signal all act through C, so they are undefined for free prototypes.

Table 10: Input-conditioned (SimPlex-GT-CS) vs. freely learned prototypes. Accuracy (%), except ROC-AUC (%) on Tolokers.

Dataset	Regime (homophily)	SimPlex-GT	Free prototypes
Wisconsin	Heterophilic (0.21)	92.16	88.64
Actor	Heterophilic (0.22)	37.68	29.36
Tolokers	Mixed (0.59)	86.17	82.86
PubMed	Homophilic (0.81)	81.42	78.50

Additive vs. gated fusion. We compare the additive fusion of Eq. 7 with a learned node-wise gate, $H'_v = (1-g_v)H_{\mathrm{loc},v} + g_v H_{\mathrm{glob},v}$, $g_v = \sigma(W_g[H_{\mathrm{loc},v} \,\|\, H_{\mathrm{glob},v}])$ (Table 11). Addition is better on Actor and Texas and the gate on Cornell, with differences inside one standard deviation. Addition keeps both branches active at every node, while their relative influence is determined by the node-dependent magnitudes produced by the two branches. An uninformative branch can therefore contribute less through a smaller output magnitude, providing soft, magnitude-driven selection without additional gating parameters. Since the explicit gate adds parameters and optimization complexity without a consistent empirical benefit, we retain simple addition in our design.

Table 11: Additive vs. gated fusion of the local and global branches. Accuracy (%).

Fusion	Actor	Texas	Cornell
Addition (ours)	37.68	92.97	84.86
Learned gate	37.23	91.62	85.40

Removing the local GCN branch. Dropping H_{loc} from Eq. 7 reduces accuracy from 92.16 to 91.56 on Wisconsin and from 84.02 to 82.20 on Cora. The larger drop on homophilic Cora matches the branch's intended role of preserving informative neighborhood structure.

K Number of Prototypes M

Table 12 shows the ablation study on the number of prototypes used in our proposed N2C attention. We tune this hyperparameter over $\{2, 5, 10, 15\}$. We can see that our proposed method is not sensitive to the number of prototypes. Although the best setting varies across datasets, usually a small number such as 5 is enough to achieve good performance while maintaining good efficiency.

Our N2C attention uses soft assignments rather than hard partitioning. Each prototype is a weighted aggregation of nodes, and each node representation is a convex combination of prototype-level

information. With soft assignments, the effect of changing M is typically gradual rather than abrupt. When M is set larger than necessary, additional prototypes do not necessarily dominate the representation; instead, they often receive relatively small effective usage or share semantic regions with other prototypes, while the node representation is still mainly determined by a subset of informative prototypes. Conversely, when M is slightly smaller, each prototype tends to cover a broader semantic scope, but the attention mechanism in Eq. 2 still allows each node to interpolate across multiple prototypes, which alleviates sharp information loss. Theorem 1 provides further intuition for this reduced sensitivity: the variance reduction factor r_c depends on the effective cluster size n_{eff}, and n_{eff} varies continuously with the soft assignment.

Table 12: The effects of the number of prototypes M in N2C

M	**Texas**	**Roman**	**Cora**
2	91.89±3.46	79.56±0.62	83.80±0.18
5	**92.97**±4.39	80.86±0.59	**84.05**±0.15
10	92.13±3.91	81.23±0.56	83.98±0.12
15	91.12±4.06	**81.56**±0.68	83.20±0.20

L Dynamic Masking Ratio r (Eqn. 10)

We further evaluate performance under different dynamic masking ratios across datasets (Table 13). While the optimal ratio varies by dataset, the best dynamic masking setting yields higher mean performance than pure random masking ($r = 0$) on heterophilic Texas ($91.89 \rightarrow 92.97$) and Roman-Empire ($80.32 \rightarrow 81.56$), while the settings perform comparably on homophilic Cora.

Table 13: The effects of r in dynamic node masking

Ratio r	**Texas**	**Roman**	**Cora**
0	91.89±3.89	80.32±0.53	84.02±0.11
0.3	90.54±3.82	81.23±0.63	83.98±0.21
0.5	91.89±3.89	**81.56**±0.68	**84.05**±0.15
0.8	**92.97**±4.39	80.96±0.57	83.50±0.12
1	92.23±3.89	79.96±0.51	83.70±0.15

M Orthogonality Regularization λ_{ORTH} (Eqn. 9)

Table 14 shows the ablation of the trade-off parameter λ_{orth}, which controls the strength of the orthogonality regularization. We can see that SimPlex-GT is not highly sensitive to this hyperparameter, but when compared with the variant without it (first row), the results show that orthogonal regularization still improves performance. This improvement can be attributed to its ability to reduce potential representation redundancy and interference, as discussed in Sec. 3.

λ_{orth} insensitivity comes from the fact that it is only an auxiliary regularizer, not the main optimization target. Its role is to reduce redundancy and interference between two branches, while the dominant learning signal still comes from the teacher–student prediction loss. Therefore, varying λ_{orth} within a moderate range (e.g., $0.3 \sim 0.5$) mainly changes the strength of branch decoupling, without altering the core task objective.

Table 14: The effects of λ_{orth}

λ_{orth}	**Texas**	**Roman**	**Cora**
0	91.89±4.05	80.23±0.55	83.50±0.11
0.3	92.54±3.66	81.12±0.53	**84.05**±0.15
0.5	**92.97**±4.39	**81.56**±0.68	83.80±0.19
0.8	92.56±4.06	80.98±0.60	83.80±0.15

N PROTOTYPE ANALYSIS

Prototype Diversity/Utilization We track two complementary metrics in Figure 3 to verify that the N2C prototypes remain diverse and balanced throughout training.

(a, c) Average Pairwise Cosine Similarity. Let $Z^{(t)} = [z_1^{(t)}, \ldots, z_M^{(t)}]^\top \in \mathbb{R}^{M \times d}$ be the prototype matrix at training step t. We compute

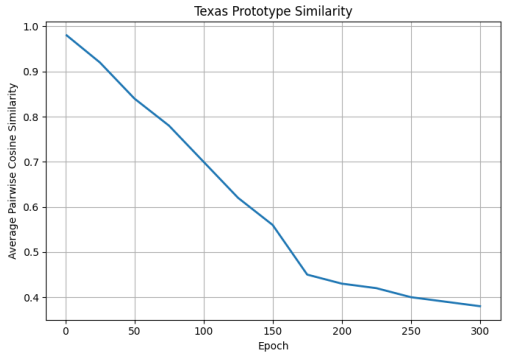
$$\mathrm{AvgCos}(t) = \frac{2}{M(M-1)} \sum_{1 \le p < q \le M} \cos(z_p^{(t)}, z_q^{(t)}), \tag{71}$$

where $\cos(a, b) = a^\top b / (\|a\|_2 \|b\|_2)$. A value close to 1 indicates prototype redundancy / near-collapse (all prototypes aligned), while decreasing AvgCos reflects growing angular diversity. On both datasets, AvgCos declines monotonically from $\sim$0.98 to $\sim$0.37 (Texas) and $\sim$0.31 (Wisconsin).

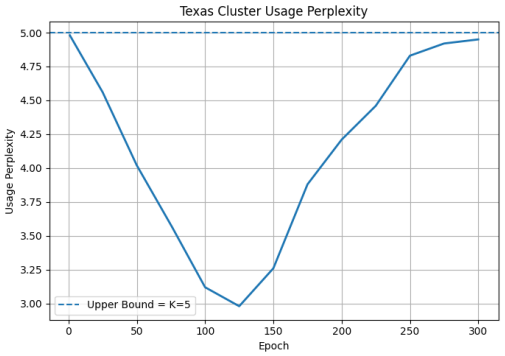
(b, d) Cluster Usage Perplexity. Let $C \in \mathbb{R}^{N \times M}$ be the soft assignment matrix and $u_k = \frac{1}{N} \sum_{i=1}^{N} C_{i,k}$ the average usage mass of prototype k, with $\sum_k u_k = 1$. The usage perplexity is

$$\mathrm{Perplexity}_u = \exp(-\sum_{k=1}^{M} u_k \log u_k) \tag{72}$$

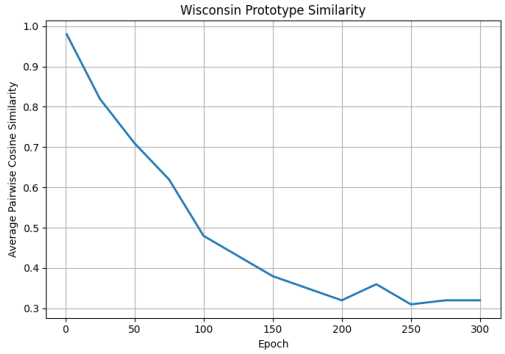
ranging from 1 (all nodes assigned to a single prototype) to M (perfectly uniform usage). Both datasets exhibit a characteristic U-shaped curve: perplexity initially drops from 5 to 3 as the model learns discriminative features, then recovers toward the upper bound M=5. This consistent trend suggests that the prototypes do not collapse into redundant representations. Instead, they evolve from initially similar and step into more diverse semantic roles, and are eventually utilized in a more balanced manner after sufficient training.



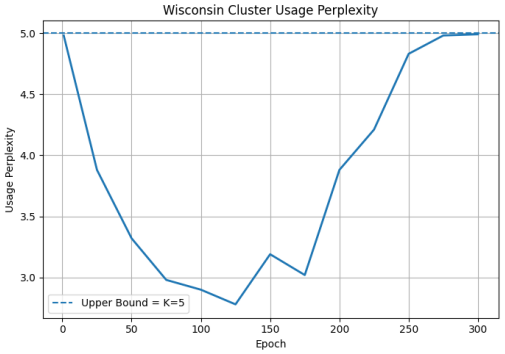
(a) Texas: Similarity



(b) Texas: Perplexity



(c) Wisconsin: Similarity



(d) Wisconsin: Perplexity

Figure 3: **Prototype stability diagnostics over training (M=5).**

Implicit Regularization We also theoretically demonstrate that our proposed model can mitigate prototype collapse or redundancy.

Let the context representation of node i in the N2C branch be

$$c_i = \sum_{k=1}^{M} a_{i,k} z_k,$$

where $a_{i,k}$ is the attention weight on prototype $z_k \in \mathbb{R}^d$, and $\sum_{k=1}^{M} a_{i,k} = 1$.

The self-supervised loss on masked nodes is

$$\mathcal{L}_{\text{N2C-SSL}} = \mathbb{E}_{i \in \mathcal{M}} \left[\| f_\theta(c_i) - T_i \|_2^2 \right],$$

where $\mathcal{M}$ is the set of masked nodes and T_i is the teacher target.

Suppose complete prototype collapse occurs, i.e.,

$$z_1 = z_2 = \cdots = z_M = z^*.$$

Then,

$$c_i = \sum_{k=1}^{M} a_{i,k} z_k = z^*, \quad \forall i,$$

so all nodes share the same context representation. The predictor then outputs a constant vector

$$\hat{y} = f_\theta(z^*),$$

and the loss reduces to

$$\mathcal{L}_{\text{N2C-SSL}}^{\text{collapsed}} = \mathbb{E}_{i \in \mathcal{M}} \left[\| \hat{y} - T_i \|_2^2 \right].$$

The optimal constant predictor is

$$\hat{y}^* = \mathbb{E}[T_i \mid i \in \mathcal{M}],$$

with minimum loss

$$\min \mathcal{L}_{\text{N2C-SSL}}^{\text{collapsed}} = \operatorname{tr}\big(\operatorname{Cov}(T_i \mid i \in \mathcal{M}) \big).$$

Therefore, as long as the teacher targets have non-zero variance, complete collapse incurs a strictly positive lower bound on the loss. This means the SSL objective itself discourages degenerate constant-prototype solutions. The argument models the N2C branch in isolation and assumes a non-collapsed teacher. In the full two-branch model, the local GCN branch varies across nodes and could in principle fit the teacher targets while the prototypes collapse, and joint collapse of student and EMA teacher would drive $\operatorname{Cov}(T_i) \to 0$. The bound therefore identifies a pressure against collapse, not a guarantee; the diagnostics in Fig. 3 are the evidence that collapse does not occur in practice.

Moreover, the EMA teacher–student framework further stabilizes training. At initialization, the student's assignment matrix

$$C = \operatorname{Softmax}(X W_{\text{cluster}})$$

is induced by randomly initialized weights, which naturally yields diverse, non-degenerate prototypes. The EMA teacher acts as a slow-moving anchor that preserves this historical diversity. Even if the student temporarily drifts toward collapse, the teacher targets T_i remain diverse, leading to a larger reconstruction loss and pulling the student back from degeneration.

O UNEVENLY DISTRIBUTED HOMOPHILIC/HETEROPHILIC

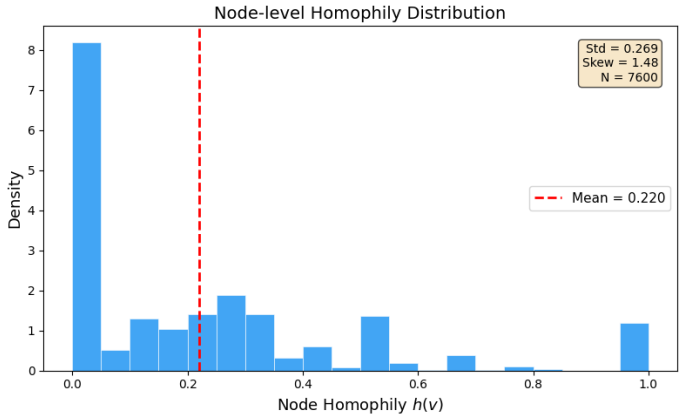
We analyze the node-level local homophily distribution $h(v)$, defined as the fraction of one-hop neighbors of node v that share the same label. We compute this metric on two representative datasets Tolokers and Actor that exhibit uneven distributions and report the statistics.

For each node v with at least one neighbor, we define its local homophily ratio as $h(v) = \frac{|\{u \in \mathcal{N}(v) : y_u = y_v\}|}{|\mathcal{N}(v)|}$, and the mean node-level homophily as $\bar{h}_{\text{node}} = \frac{1}{N} \sum_v h(v)$. **Note:** It differs from the edge homophily ratio $h_{\text{edge}} = |\{(u,v) \in E : y_u = y_v\}| / |E|$ reported in our submission. The y-axis shows the probability density of node-level homophily values, where for a histogram bin $B = [a, b)$, the plotted height is $\text{Density}(B) = \frac{|\{v : h(v) \in B\}|}{N\,(b-a)}$ with N denoting the number of valid nodes; hence the total area under the histogram satisfies $\sum_B \text{Density}(B) \cdot (b - a) = 1$

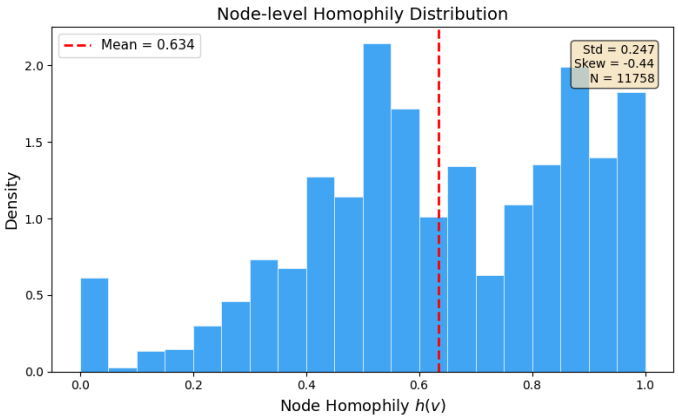
(a) Actor is globally heterophilic ($\bar{h}_{\text{node}} = 0.220$), yet its distribution is highly uneven: 55.1% of nodes fall in the strong-heterophily bin $h(v) \in [0, 0.2)$, while 6.1% reside in the strong-homophily bin $h(v) \in [0.8, 1.0]$, yielding a right-skewed distribution (Skew$= 1.48$, Std$= 0.269$).

(b) Tolokers is globally homophilic ($\bar{h}_{\text{node}} = 0.634$), but its node-level homophily is broadly dispersed across the entire $[0, 1]$ range: 32.8% of nodes are strongly homophilic ($h(v) \in [0.8, 1.0]$), yet 15.5% are heterophilic ($h(v) < 0.4$), with a mild left skew (Skew$= -0.44$, Std$= 0.247$). In both cases, a single global homophily ratio fails to capture the substantial heterogeneity across nodes. These results support our claim that the local homophily patterns in such datasets are *unevenly distributed across nodes*.

Overall, these results suggest that our evaluation already covers graphs with markedly uneven local patterns. Under these settings, SimPlex-GT still achieves strong performance, reaching 86.17% on Tolokers in supplementary experiments and 37.68% on Actor, both outperforming the baselines.



(a) Actor ($\bar{h}_{\text{node}} = 0.220$, Skew$= 1.48$)



(b) Tolokers ($\bar{h}_{\text{node}} = 0.634$, Skew$= -0.44$)

Figure 4: **Node-level homophily distributions.**

P Dataset Statistics

We summarize the details of the datasets used in our experiments in Table 16.

Q Hyperparameters

We provide the major hyperparameter search space:

- Number of clusters: $\{2,\ 5,\ 10,\ 15\}$.

Table 15: **Node classification ROC-AUC (%) on Tolokers.** Results are reported as mean ± std over 10 official splits from (Platonov et al., 2023). Tolokers serves here as a representative case of a graph with markedly uneven node-level homophily.

Category	Model	ROC-AUC (%)
Classic GNNs	GCN	83.64 ± 0.67
	GraphSAGE	82.43 ± 0.44
	GAT	83.70 ± 0.47
Heterophily GNNs	H2GCN	73.35 ± 1.01
	GPRGNN	72.94 ± 0.97
	GloGNN	73.39 ± 1.17
	FAGCN	77.75 ± 1.05
	S3GCL	79.39 ± 0.44
	GREET	79.21 ± 0.79
Graph Transformers	GT	83.23 ± 0.64
	NAGphormer	79.32 ± 0.39
	NodeFormer	77.98 ± 0.95
	GOAT	83.11 ± 1.04
Ours	SIMPLEX-GT-CS	**86.17** ± 0.62
	SIMPLEX-GT-CF	**86.32** ± 0.46

Table 16: Dataset statistics. "Homo" denotes the homophily ratio.

Dataset	Nodes	Edges	Features	Classes	Homo
Cornell	183	295	1,703	5	0.30
Texas	183	309	1,703	5	0.11
Wisconsin	251	499	1,703	5	0.21
Actor	7,600	29,926	932	5	0.22
Chameleon (Filtered)	890	17,708	2,325	5	0.24
Squirrel (Filtered)	2,223	93,996	2,089	5	0.21
Roman-Empire	22,662	32,927	300	18	0.05
Cora	2,708	10,556	1,433	7	0.81
CiteSeer	3,327	9,104	3,703	6	0.74
PubMed	19,717	88,648	500	3	0.80
Ogbn-Arxiv	169,343	1,166,243	128	40	0.66
Penn94	41,554	1,362,229	5	2	0.47
Tolokers	11,758	519,000	10	2	0.59
Amazon-Ratings	24,492	93,050	300	5	0.38

- λ_{orth}: $\{0,\ 0.1,\ 0.3,\ 0.5,\ 0.8\}$.
- Learning rate for Encoder: $\{0.01,\ 0.005,\ 0.001\}$.
- Dropout for GCN: $\{0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.8\}$.
- Dropout for Attention: $\{0.1,\ 0.3,\ 0.5,\ 0.7,\ 0.8\}$.
- Dimension of tokens: $\{128,\ 256,\ 512,\ 1024,\ 2048,\ 4096\}$.
- Total masking ratio: $\{0.9,\ 0.8,\ 0.5,\ 0.3,\ 0.1,\ 0\}$.
- Dynamic masking ratio: $\{0.9,\ 0.8,\ 0.5,\ 0.3,\ 0.1,\ 0\}$.
- Momentum: $\{0.9,\ 0.99,\ 0.999\}$.