Node2Graph

Diagnosing Task Unification in Graph Learning

Neelam Akula   Varun Shiralkar   Baris Coskunuzer

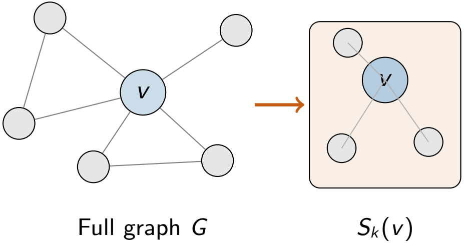
Department of Mathematical Sciences
University of Texas at Dallas

PAKDD 2026

Graph Foundation Models & Task Unification

The appeal of task unification:

- Graph foundation models serve *many tasks* with one backbone
- Popular strategy: cast **node classification** as **subgraph classification**
  - Extract k-hop ego subgraph $S_k(v)$ around anchor v
  - Classify the subgraph $\Rightarrow$ predict label y_v
- Enables shared pretraining



The open question:

When is the node→subgraph conversion faithful,
and when does it quietly lose the signal that matters?

Problem Setup: Node → Graph Task Transfer

Task Transfer Operator T_k

Given $G = (V, E, X, Y)$, define the k-hop ego subgraph with anchor flag + radial distance encoding:

$$S_k(v) = G\left[\mathcal{N}_k(v) \cup \{v\}\right]$$

Node task: $f_{\text{node}} : (G, v) \mapsto \hat{y}_v$

Subgraph task: $f_{\text{sub}} : S_k(v) \mapsto \hat{y}_v$

Transfer Gap:

$$\Delta_k = \mathrm{Acc}_{\text{node}} - \mathrm{Acc}_{\text{sub}}(k)$$

Evaluated for $k \in \{1, 2, 3\}$ across 9 benchmarks & 6 backbones.

Experimental controls:

- Same backbone & hyperparameter grid
- Leakage-free splits via anchor grouping
- Compute-matched training budgets
- Mean / attention / sort pooling

Alignment Metric: N2GLA

Key insight: Transfer quality depends on how *label-supportive* the k-hop neighborhood is.

k-Hop Graph Label Alignment ($\mathrm{N2GLA}_k$)

For node v with label y_v, fraction of same-class nodes in its ego:

$$\mathrm{N2GLA}_k(v) = \frac{1 + \left|\{u \in \mathcal{N}_k(v) : y_u = y_v\}\right|}{1 + \left|\mathcal{N}_k(v)\right|} \qquad \mathrm{N2GLA}_k = \frac{1}{|V|} \sum_{v \in V} \mathrm{N2GLA}_k(v)$$

Properties:

- *Anchor-centric* — computed on the exact ego the classifier sees
- Distinct from global assortativity or edge homophily
- Label-based: usable in semi-supervised settings
- Training-free: cheap to compute upfront

Prediction

$\mathrm{N2GLA}_k$ **high** $\Rightarrow$ safe transfer

$\mathrm{N2GLA}_k$ **low** $\Rightarrow$ large gap, worsens with k

Experimental Setup

Datasets (9 total):

Dataset	Node Hom.	Type
Cora	0.83	Citation
Citeseer	0.72	Citation
PubMed	0.79	Citation
Texas	0.10	Webpage
Cornell	0.39	Webpage
Wisconsin	0.15	Webpage
Actor	0.20	Co-occ.
Roman-Emp.	0.05	Wikipedia
Chameleon*	0.24	Wikipedia

*Filtered variant

Backbones (6):

- GNNs: GCN, GraphSAGE, GAT
- Graph Transformers: GPS, GT
- Propagation: GPR-GNN

Protocol:

- 60/20/20 train/val/test split
- 5 seeded runs, mean accuracy reported
- $k \in \{1, 2, 3\}$ hop subgraphs
- Adam, cosine LR decay, early stopping

Results: Homophilic Graphs — Transfer is Safe

RQ1: Subgraph classification *closely tracks* node classification on homophilic graphs.

Backbone	Task	Cora	Citeseer	PubMed
GCN	NC	88.18	76.04	87.94
	1-hop	84.57	71.92	86.76
	2-hop	82.87	70.75	81.89
	3-hop	81.40	69.85	80.93
GSAGE	NC	87.33	75.05	89.53
	1-hop	85.12	71.23	87.58
	2-hop	82.54	70.51	81.54
	3-hop	81.95	69.22	81.20

Finding (RQ1): Gaps at $k=1$ are **small ($\sim$1–4%)** and grow only mildly with k. Consistent across all six backbones — the bottleneck is the conversion, not the encoder.

Results: Heterophilic Graphs — Transfer Breaks Down

RQ2: Gaps are large, grow with k, and are backbone-invariant.

Backbone	Task	Texas	Cornell	Wisconsin	Actor	Roman
GSAGE	NC	81.05	82.63	85.88	35.05	78.08
	1-hop	42.11	35.26	41.57	27.30	68.27
	2-hop	36.32	25.79	37.65	25.08	37.21
	3-hop	35.79	33.68	41.96	23.95	14.01
GT	NC	71.05	65.26	73.33	36.57	74.91
	1-hop	39.47	37.37	33.33	28.87	67.00
	2-hop	29.47	28.95	29.02	26.28	30.15
	3-hop	24.74	31.05	31.37	25.70	13.13

Accuracy drops reach **30–65%** on Roman-Empire, Texas, Cornell. GNNs and graph transformers **fail equally** — the cause is the conversion, not the encoder.

N2GLA Predicts the Transfer Gap

Alignment scores ($\mathrm{N2GLA}_k$):

Dataset	1-hop	2-hop	3-hop
Cora	0.871	0.775	0.663
Citeseer	0.820	0.758	0.712
PubMed	0.865	0.761	0.687
Texas	0.391	0.535	0.354
Cornell	0.427	0.408	0.289
Wisconsin	0.422	0.434	0.327
Actor	0.412	0.240	0.216
Roman	0.305	0.193	0.147
Chameleon*	0.350	0.296	0.245
Corr. w/ Hom.	**0.97**	**0.87**	**0.92**

Avg. gap vs. $\mathrm{N2GLA}_k$ (Pearson r):

	Corr. w/ $\mathrm{N2GLA}_k$
Δ_1	-0.52
Δ_2	-0.47
Δ_3	-0.60

Key takeaways:

- $\mathrm{N2GLA}_k$ strongly tracks homophily ($r \geq 0.87$)
- Higher $\mathrm{N2GLA}_k \Rightarrow$ smaller transfer gap
- On heterophilic graphs, $\mathrm{N2GLA}_k$ falls as k grows, amplifying the gap
- Occasional $k{=}2$ bump (e.g. Texas) before $k{=}3$ drop

Transfer Gap vs. Alignment: Backbone-Invariant Pattern

avg.pdf

What the plot shows:

- Each point = one dataset at one k
- **Triangles**: homophilic — high alignment, low gap
- **Squares**: heterophilic — low alignment, high gap
- Increasing k moves heterophilic points *right* (lower $\mathrm{N2GLA}_k$) and *up* (larger Δ_k)
- Pattern holds across **all 6 backbones**

$\mathrm{N2GLA}_k$ is a **training-free diagnostic**: compute it before any model training to predict transfer fidelity.

Practical Guidance: Alignment-Aware Unification (RQ3)

Decision rule for node→subgraph:

1. Compute $\mathrm{N2GLA}_k$ for small k on training graph

(i) $\mathrm{N2GLA}_1$ high ⇒ use $k{=}1$; interface is **safe**

(ii) $\mathrm{N2GLA}_k$ low for all k ⇒ **avoid naïve reformulation**

(iii) Apply **alignment-aware variants** to narrow gap

Alignment-aware variants:

- **Outer-shell downweighting** — reduce distant-neighbor influence
- **Boundary normalization** — normalize at each hop boundary
- **Relation-aware channels** — separate same- vs. cross-class edges

$\mathrm{N2GLA}_1$ at a glance

Dataset	$\mathrm{N2GLA}_1$
Cora	0.871
PubMed	0.865
Citeseer	0.820
Roman	0.305
Texas	0.391
Actor	0.412
Wisconsin	0.422

These fixes reduce heterophilic gaps **without breaking the unified interface.**

Conclusion

Question: When does node$\rightarrow$ k-hop subgraph conversion preserve accuracy?

Findings:

- **Homophilic:** near parity at k=1; mild, monotone drops. **Safe to unify.**
- **Heterophilic:** gaps up to 65%, growing with k. **Backbone-invariant.**
- **N2GLA$_k$** reliably predicts the gap ($r \geq 0.87$ with homophily; negative correlation with Δ_k).
- Alignment-aware pooling & boundary normalization narrow the gap while keeping the unified interface.

Practical Takeaway

High $\mathrm{N2GLA}_1$ $\Rightarrow$ safe to unify.

Low $\mathrm{N2GLA}_k$ $\Rightarrow$ use alignment-aware pooling or avoid subgraph reformulation.

Future work:

- Alignment-conditioned pretraining
- Adaptive k selection during training
- Formal guarantees for N2GLA
- Link prediction, temporal & multi-relational graphs

Code: `https://anonymous.4open.science/r/node2graph-D112`

Thank You!

Questions?

{neelam.akula, varun.shiralkar, coskunuz}@utdallas.edu

Supported by NSF DMS-2220613 and DMS-2229417.
Computational resources: TACC at UT Austin.