ICML 2026 · SEOUL

Same Graph Cross-Task Transfer in GNNs

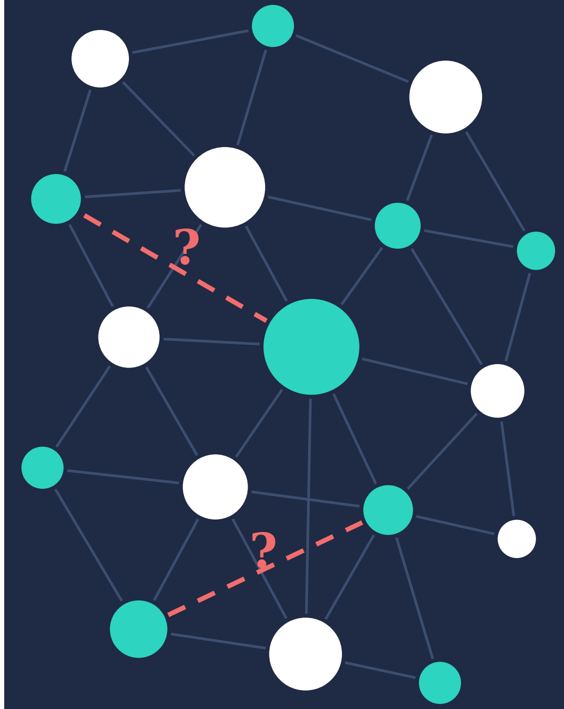
Protocols and Predictors

When does reusing supervision across node classification and link prediction help on the same graph?

Neelam Akula[*1] **Surbhi Kumar**[*1] **Murat Kantarcioglu**[2] **Baris Coskunuzer**[1]

[1] University of Texas at Dallas [2] Virginia Tech [*] Equal contribution

Proceedings of the 43rd International Conference on Machine Learning, Seoul, 2026.



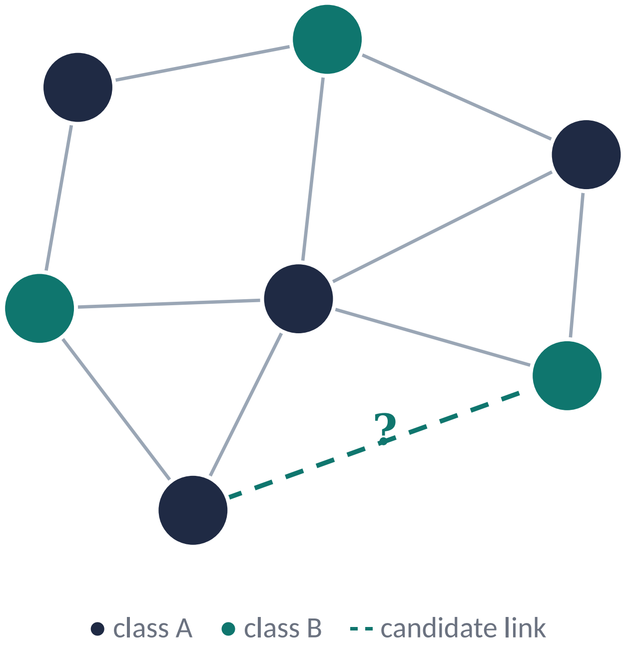
ICML 2026

Multiple tasks on one graph

Real systems answer several questions over the same underlying graph. The supervision for both tasks already coexists in the data: node labels for **node classification (NC)** and observed edges for **link prediction (LP)**.

Example (social network). NC asks whether a node is a human user or a bot; LP suggests potential friendship connections.

Two practical needs. Can supervision from one task improve the other, and can a single shared encoder serve both to amortize training and serving cost?



Contributions

1. **Problem setting: same graph cross-task learning.** We formalize NC and LP reuse on one underlying graph and study both directions, NC→LP and LP→NC.

2. **Leakage-free evaluation protocol.** A standardized setup that fixes splits, observed-graph construction, and LP negatives, preventing leakage from evaluation edges.

3. **Systematic study across backbones and transfer strategies.** Five transfer strategies (WS, ET-Rep, ET-Concat, MV, Joint) over three backbones (GCN, GraphSAGE, GPS); a robust directional asymmetry across architectures.

4. **Multi-objective model selection and predictors.** CoTask Score (CTS) summarizes joint NC+LP utility; homophily predicts NC→LP gains, while LP→NC gains concentrate in structure-dominant regimes.

Leakage-free evaluation protocol

A valid cross-task comparison must satisfy three requirements, applied identically to NC, LP, and transfer.

i **No edge leakage.** Edges evaluated for LP must not appear in the graph used for message passing.

ii **Fixed negatives.** LP uses fixed negative sets for train, validation, and test, reused across all methods.

iii **Fixed splits.** All methods use the same node split for NC and the same edge split for LP positives.

Setup. Observed adjacency $A_{obs} = \mathrm{Adj}(V, E_{+tr})$. Splits 60/20/20 (NC) and 80/10/10 (LP); negatives at 1:1, fixed seeds, reused across methods.

Cross-task transfer as optimization

For a fixed backbone, NC and LP define different losses over the same parameters. We study five lightweight regimes:

- **WS:** initialize the target encoder from a source-task solution.
- **ET-Rep / ET-Concat:** reuse frozen source embeddings as input (replace or concatenate).
- **MV:** two encoders trained jointly with cross-view alignment.
- **Joint:** one shared encoder, two heads, both losses.

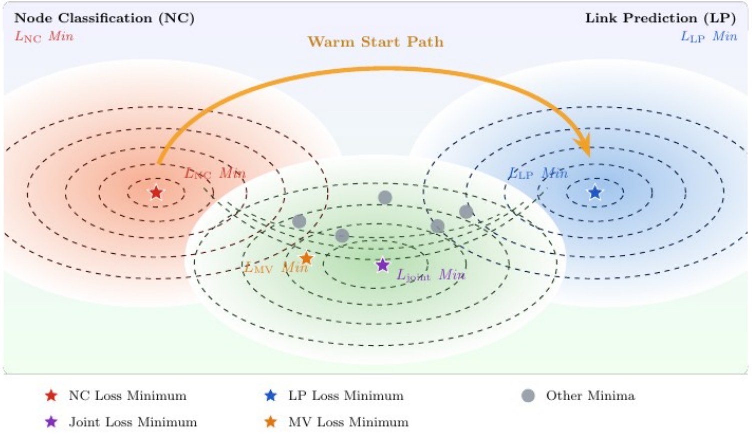


Figure 1. NC and LP are different losses over the same parameters, with possibly incompatible minima.

A directional asymmetry (GCN)

	NC→LP ΔAUC (pp)						LP→NC ΔAcc (pp)					
Dataset	Base	WS	ET-R	ET-C	MV	Joint	Base	WS	ET-R	ET-C	MV	Joint
Homophilic												
Cora	80.3	+9.4	+10.7	+9.8	+10.1	**+12.3**	85.3	**+0.6**	-12.7	+0.5	+0.5	+0.4
Citeseer	77.0	+11.9	+11.9	+11.1	+12.8	**+13.2**	71.6	+0.5	-16.8	+0.1	+0.3	**+3.5**
PubMed	89.8	**+5.9**	+4.1	+4.5	+5.0	+4.7	88.5	**+0.1**	-21.4	-1.0	-0.1	-0.5
Heterophilic												
Texas	67.0	+3.6	-0.3	-2.6	+4.7	**+7.1**	50.3	+2.6	+6.6	+1.6	-0.3	**+11.9**
Cornell	75.4	**+4.6**	-2.1	-1.3	+3.7	+2.3	51.6	**-0.3**	-11.1	-2.9	-2.4	-2.6
Wisconsin	75.1	+1.8	-0.2	-0.4	+2.3	**+3.7**	49.4	-0.2	-2.9	-1.6	-0.8	**+1.0**
Actor	80.4	-0.3	-1.6	-0.7	-0.3	**+1.2**	27.9	+0.5	-1.3	+1.0	+0.7	**+3.2**
Roman	73.8	**+0.1**	-10.6	-6.6	-0.1	-11.1	51.4	**+0.2**	-28.2	-0.4	0.0	-0.5
Structure-dominant												
USA	95.5	**+0.3**	+0.1	+0.3	+0.3	-0.6	54.6	-1.1	+5.8	**+7.8**	-0.2	+6.2
Europe	92.6	**+0.4**	-1.1	+0.1	+0.2	-2.3	53.7	-0.4	+2.6	+1.4	-1.0	**+2.8**
Brazil	90.4	**+3.1**	+0.1	+0.9	+1.3	+0.1	48.9	-0.4	**+15.2**	+6.7	+2.2	+13.7

GCN backbone. Gains in percentage points vs the single-task base (Base = base score). Bold = best of the five mechanisms per direction. ET-R = ET-Rep, ET-C = ET-Concat.

Homophily explains the asymmetry

NC→LP gains correlate with homophily (strongest for ET-Concat and ET-Rep), while LP→NC is not. NC training amplifies the cosine-similarity gap between linked and unlinked pairs, about 2.1x more on homophilic graphs **(r = +0.901).**

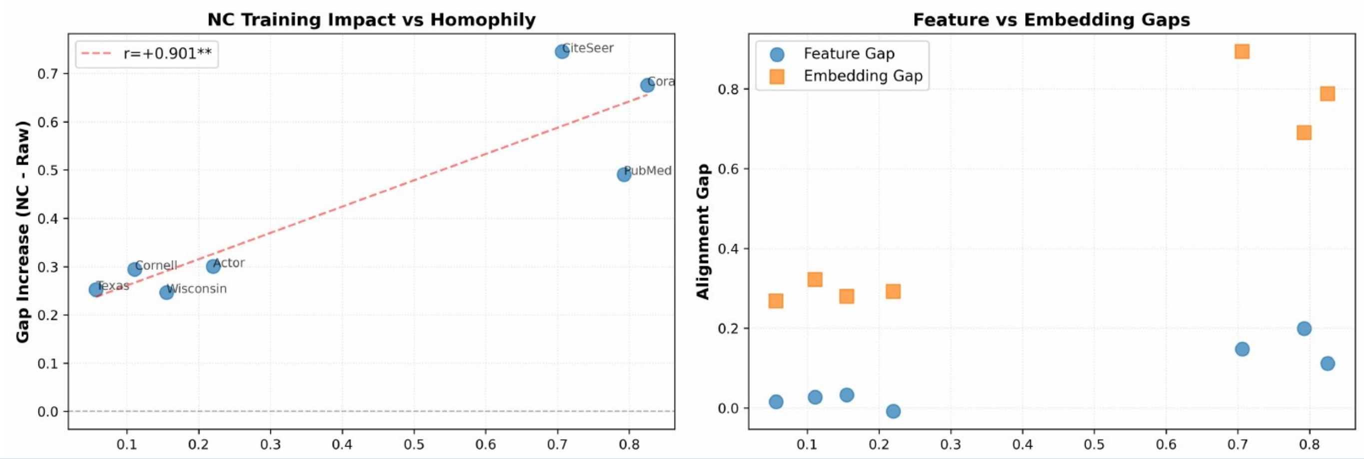


Figure 2. NC-training cosine-gap increase vs. node homophily (left); NC-trained embeddings vs. raw-feature gaps (right).

Conclusion

- Same graph cross-task transfer is strongly directional and predictable.
- NC→LP is reliably beneficial on homophilic graphs. LP→NC is less predictable and helps mainly in structure-dominant regimes.
- Coupled training (MV, Joint) is typically more stable than post hoc representation reuse.
- Homophily explains NC→LP gains more consistently than LP→NC.

Practical guidance. Homophilic graph: prefer NC→LP with a coupled objective (MV or Joint). Structure-dominant (LP easy, NC unsaturated): LP→NC can help. Avoid naive ET-Rep reuse.

Code & data: **github.com/avp-neelam/CrossTaskTransfer**