Can Graph Foundation Models Generalize Over Architecture?

Benjamin Gutteridge, Michael Bronstein, Xiaowen Dong, University of Oxford

TL;DR

Existing domain-agnostic GFMs rely on **fixed architectural backbones** and are provably limited to short-range tasks. We introduce **GOBLIN**, an architecture-adaptive mixture-of-experts which discovers GNN operators at inference time, enabling zero-shot generalization across tasks with heterogeneous architectural requirements, including **long-range tasks**.

Background: Domain-Agnostic GFMs

We study **zero-shot transductive node classification** (label inpainting): given (A, X, Y_L), predict labels for $V_U = V \setminus V_L$ without retraining on the target graph. Prior GFMs:

- **GraphAny** — per-node MoE over a **fixed** basis of linear GNNs $\hat{Y} = SXW$ (analytic solution $W^* = (SX)^+_L Y_L$), with operator set $\mathcal{S} = \{I,\ A,\ A^2,\ (I-A),\ (I-A)^2\}$ (1-, 2-hop linear GNNs)
- **TS-GNN** — enforces node, label, feature symmetry via DeepSets; **fixed** MPNN backbone.

Both generalize well across diverse real-world benchmarks — but those benchmarks are **all short-range tasks**. Both methods use $\leq$ **2** message-passing layers $\rightarrow$ *implicit architectural overfitting*.

Range as a Diagnostic Axis

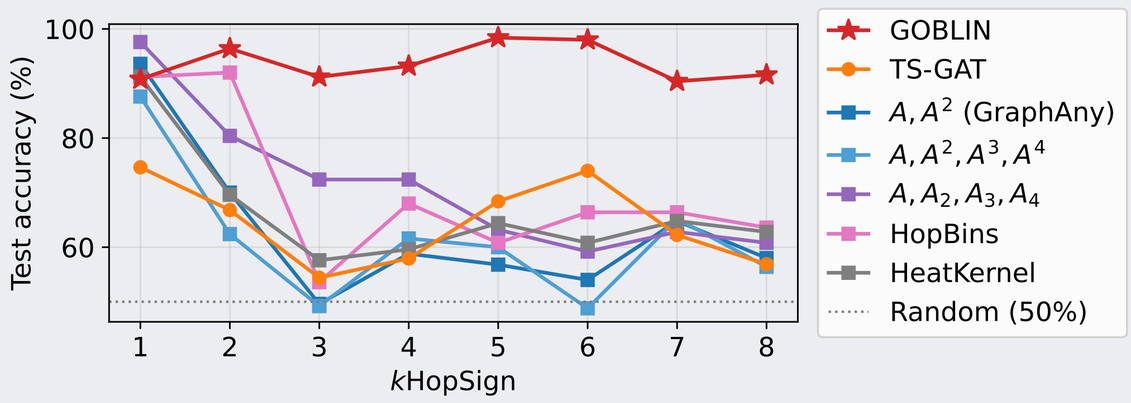
Range ρ_u (Bamberger et al., 2025) measures distance-weighted pairwise node sensitivity. For a linear GNN $\hat{Y} = SXW$:

$$\rho_u = \frac{\sum_v |S_{uv}|\, d_G(u, v)}{\sum_v |S_{uv}|}.$$

Range depends only on S and graph topology; GraphAny standard operators have range $\leq$ **2** hops, TS-GNNs also fixed by $\rho <$ depth; **GraphAny/TS-GNN cannot solve range $>$ 2 tasks.**

Synthetic Varying-Range Task; Fixed Architectures

Synthetic task with controllable range $\approx k$: label is the sign of summed $\mathcal{N}(0, 1)$ features at graph distance $\approx k$. Various GraphAny operator bases trained on Cora, applied zero-shot.



All fixed bases collapse to near-random for $k > 2$, even when expanded (A^3, A^4, precise-hop A_k). Moreover, expanding the basis to help on kHopSign **degrades standard benchmark performance** (all Δ below are negative):

Operators	Avg. benchmark Δ%
A, A^2 (GraphAny)	**0.0**
A, A^2, A^3, A^4	**−5.17**
A, A_2, A_3, A_4	**−4.95**
HopBins	**−5.29**
HeatKernel	**−3.02**

$\rightarrow$ *No one-size-fits-all architecture.* **Need an adaptive basis.**

GOBLIN: Inference-Time Architecture Adaptation

GOBLIN discovers a task-appropriate operator basis at inference time from labeled nodes V_L only — no retraining, no backpropagation on V_U.

Parameterized operator family:

- $S^{\text{Gauss}}(\mu, \sigma)$: soft aggregation at geodesic distance $\approx \mu$ (collapses to exact k-hop when $\sigma = 0$). Approximates multi-hop, precise-hop.
- $S^{\text{Heat}}(\tau) = \exp(-\tau L_{\text{sym}})$: heat-kernel diffusion with distance range $\propto \sqrt{\tau}$. Approximates multi-layer MPNNs/low-pass.

Together these span diverse architectural axes, and have intrinsic geodesic-/resistance-derived notions of range.

Stage 1 — Basis search. For each candidate θ, fit a linear GNN expert $\hat{Y}^\theta = S(\theta)X(S(\theta)X_L)^+ Y_L$ and score on a held-out split of labeled nodes. Bayesian optimization (UCB) searches θ; then $K = 4$ diverse operators are selected greedily.

Stage 2 — Permutation-invariant MoE. A **DeepSet** over per-operator disagreement features assigns per-node weights $\alpha_u^{(i)}$: $\hat{Y}_U = \sum_i \alpha_u^{(i)} \hat{Y}_U^{(i)}$. Invariant to operator ordering and basis size; DeepSet MLP trained once on source graph, applied zero-shot.

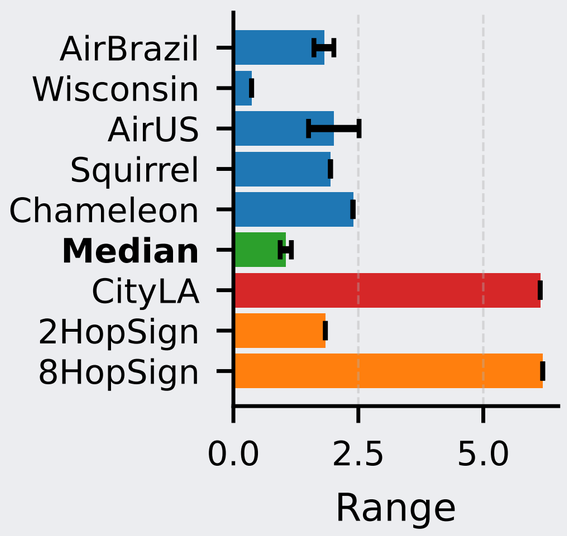
Experiments

GAT trained end-to-end per graph; easier setting. GFMs trained on Cora, eval zero-shot. Avg. % accuracy across benchmark sets.

Benchmarks (#Graphs)	GAT	GraphAny	TS-GNN	**GOBLIN**
Short-range (25)	65.98	65.76	66.20	**68.03**
Long-range (4)	20.35	18.53	17.44	**23.50**

GOBLIN $\sim$**2%** above next-best on short-range; gains tend to be on **longer-range tasks**. Fixed-backbone GFMs under-reach; $\sim$ **3–8%** lower performance on long-range CityNetworks.

GOBLIN adapts to task range: inferred operator ranges increase with task range.



Benchmarks with highest **Δ%** Acc. over competitors tend to be **longer-range**; 4/5 much higher than 25-benchmark avg.

GOBLIN finds **long-range operators** for long-range tasks (and vice-versa for short)

Conclusion

- No one-size-fits-all GNN architecture exists $\rightarrow$ **architecture adaptivity** necessary for GFMs.
- GOBLIN $\rightarrow$ GNN MoE w/ inference-time operator discovery — outperforms fixed-backbone GFMs on short-/long-range benchmarks.
- Future GFM eval should include **architecturally diverse** tasks; standard benchmarks mask implicit architectural overfitting.

beng@robots.ox.ac.uk

GRaM Workshop @ ICLR 2026

github.com/BenGutteridge/GOBLIN