Overcoming Information Bottlenecks in Directed Graph Neural Networks through Rewiring

Yu He, Ishani Karmarkar, Ellen Vitercik   Stanford University

TL;DR: We introduce a rewiring framework for directed graphs that leverages **shortcut sets** to address long-range dependencies and **edge connectivity** to alleviate oversquashing, while preserving the graph's inherent asymmetries and reachability constraints.

Directed GNNs

- Dir-GNN [1]: directed message passing (in, out edges)
- Information bottlenecks
  - Long-range dependency
  - Oversquashing
- Most prior rewiring strategies focus on undirected graphs and many are based on spectral theory, which is less well-defined for directed graphs.

Shortcut set-based rewiring

d-Shortcut Set

Given a directed graph $G = (V, E)$, a d-shortcut set is a set of edges S such that the augmented graph $H = (V, E \cup S)$ preserves the same reachability as G, and for every reachable pair $u, v \in V$, there exists a path from u to v in H of length at most d.

$\sqrt{n}$-sampling algorithm

Guarantees $d = O(\sqrt{n})$ with $O(n)$ additional edges:

- Sample a landmark set $L \subseteq V$ with node probability $1/\sqrt{n}$.
- Construct a complete graph $G' = (L, L \times L)$.
- For each edge $(a, b) \in L \times L$, if they are reachable in G, then we add (a, b) to the shortcut set S.

Shortcut set-based rewiring

- Reduces diameter to tackle long-range dependency.
- Preserve reachability constraints to model asymmetries.
- The linear edge budget ensures efficiency.

Edge connectivity-based rewiring

Edge connectivity

Let $\lambda(u, v)$ be the edge connectivity from node u to node v,

$$\lambda(u, v) = \min_{E' \subseteq E}\{\,|E'| : u \not\to v \text{ in } G = (V, E \setminus E')\}$$

In words: the minimum number of edges whose removal destroys all directed paths from u to v.

Scalable version: l-hop edge connectivity

- Given a budget k edges to rewire, we sample $K \gg k$ pairs of nodes $u, v \in V$.
- Compute the l-hop neighborhood of u, where l is number of MPNN layers, to preserve locality.
- Retain u, v where v is reachable from u.
- Compute the edge connectivity of (u, v) within the l-hop neighborhood.
- Choose k pairs with lowest edge connectivity to rewire.

Edge connectivity-based rewiring

- Edge connectivity directly targets bottlenecks, as it measures the number of independent information paths.
- Low connectivity indicates higher risk of oversquashing.
- Rewire within l-hop neighborhoods for improving scalability and preserving locality.

Relation types

Rewiring augments the graph with new edges that play different roles: original edges preserve structural information, while rewired edges improve connectivity.

Relational GNNs (R-GNNs) [2] assign separate parameters to different edge types:

$$\mathbf{h}_v^{(k+1)} = \phi_k\Big(\mathbf{h}_v^{(k)}, \sum_{r \in \mathcal{R}} \sum_{u \in \mathcal{N}_r(v)} \psi_{k,r}\Big(\mathbf{h}_u^{(k)}, \mathbf{h}_v^{(k)}\Big)\Big),$$

where $r \in \mathscr{R}$ denotes a relation type, and we assign $r = 0$ to original edges and $r = 1$ to rewired edges.

Experiments

Synthetic experiments

We design three tasks to specifically target long-range dependency and oversquashing in directed graphs:

- Cycle detection (Cycle)
- Reachability (Reach.)
- Topological sort (Topo.)

Table 1: Performance of rewiring strategies on synthetic datasets (mean ± std). Best in red, second best in blue.

Method	Cycle	Reach.	Topo.
None	68.46 ± 1.65	71.53 ± 6.93	2.75 ± 0.64
SDRF	68.18 ± 2.11	74.01 ± 4.93	2.51 ± 1.04
Shortcut	68.29 ± 2.18	75.28 ± 8.85	3.01 ± 1.09
Greedy EC	69.09 ± 3.10	77.12 ± 7.23	3.16 ± 0.45

Real-world experiments

We evaluate on two real-world heterophilic webpage graphs (Chameleon, Squirrel). We also include a Combined strategy that applies shortcut set and edge connectivity-based rewiring jointly.

Table 2: Performance of different rewiring strategies on real-world datasets (mean ± std). The best is highlighted in red and the second best is highlighted in blue.

Dataset	Model	None	SDRF	Shortcut Set	Greedy EC	Combined
Chameleon	Dir-GCN	78.88 ± 2.25	79.17 ± 1.55	79.60 ± 1.36	79.23 ± 2.13	78.96 ± 1.69
	Dir-SAGE	58.37 ± 1.94	58.91 ± 1.91	58.93 ± 1.79	59.56 ± 1.91	58.29 ± 1.19
	Dir-GAT	63.33 ± 2.17	64.32 ± 2.31	63.46 ± 2.10	64.91 ± 3.34	63.77 ± 3.04
Squirrel	Dir-GCN	73.45 ± 1.25	74.23 ± 1.19	74.59 ± 1.82	75.02 ± 1.58	74.78 ± 1.92
	Dir-SAGE	45.92 ± 2.02	46.09 ± 2.98	46.74 ± 1.99	46.54 ± 2.63	46.42 ± 2.70
	Dir-GAT	65.82 ± 2.90	66.34 ± 2.31	65.92 ± 2.42	66.52 ± 2.19	66.32 ± 2.21

Outlook

- Many real-world systems inherently require asymmetry.
- Directionality remains underexplored in graph learning, with many open questions.
- Need for dedicated benchmarks on directed graphs addressing long-range dependency and oversquashing.
- Need for better architectures for rewiring that adapt between original and modified edges.

[1] Rossi et al., Edge directionality improves learning on heterophilic graphs. LoG 2023.
[2] Battaglia et al., Relational inductive biases, deep learning, and graph networks. 2018.

Full paper