NEURAL INFORMATION PROCESSING SYSTEMS 2026

Revisiting Autoregressive GCNs for Vehicle Routing Problems

Zhipeng Zhong*, Junquan Huang*, Yu Huang, Boyuan Zheng, Jungang Li, Zihao Dongfang, Song Dai, Xuming Hu, and Zong-Gan Chen

***Equal contribution**

Presenter: Junquan Huang

South China Normal University

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Static heatmaps and dynamic routing

The next feasible stop depends on the route so far.

Conventional GCN

Score the edges once

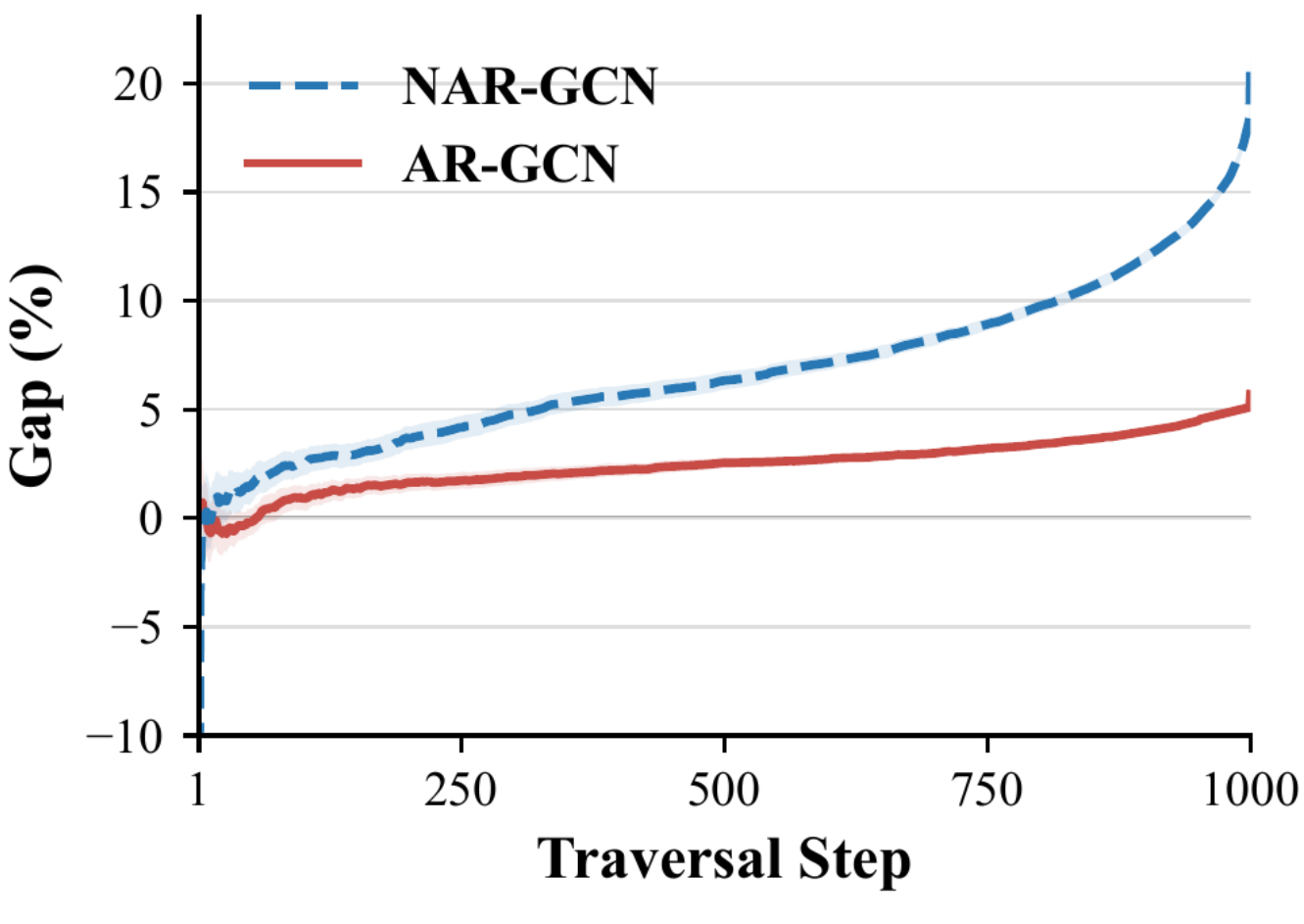
A static heatmap guides the search that builds or refines the route.

Our AR-GCN

Re-encode at each step

The GCN scores the current subgraph again before choosing the next feasible stop.

The gap widens late with a static heatmap



Symmetric TSP with 1,000 nodes (Figure 1)

Greedy decisions on fixed scores

The dashed line rises most sharply in the later stages. The AR-GCN curve grows more gradually.

Final gap on uniform STSP-10K

19.46% vs. 6.12%

Same backbone, NAR-GCN vs. AR-GCN

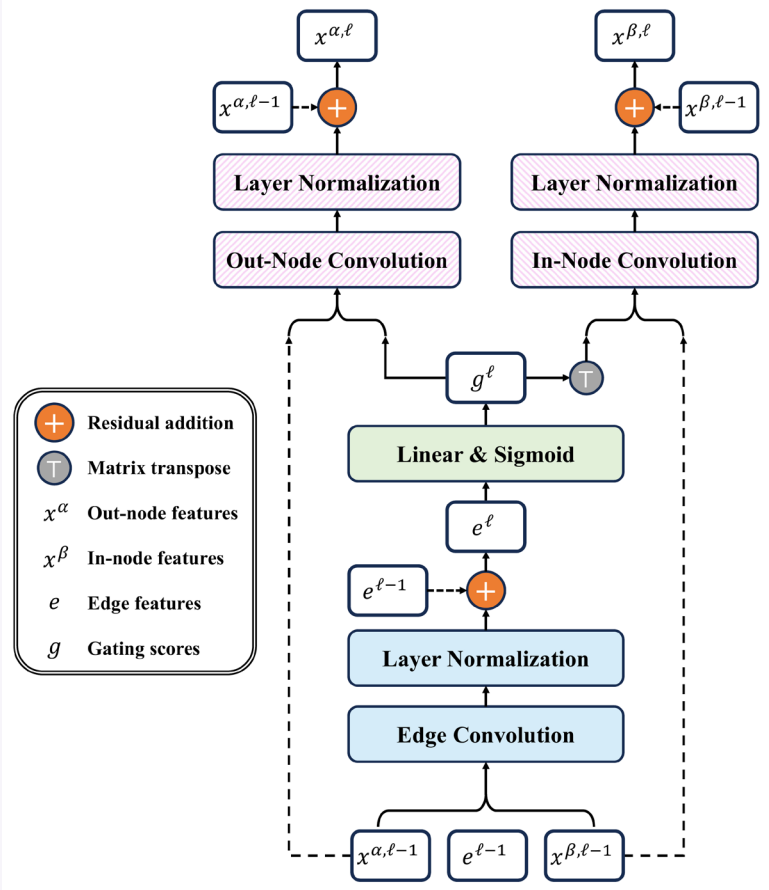
Incoming and outgoing edges play different roles

The cost from A to B can differ from the cost from B to A.

Dual node representations keep incoming and outgoing signals separate. Directed edge features control each message.

6.75% vs. 1.64%

Mean gap on ATSP-1K, single vs. dual representation



Dual-node gated GCN layer (Figure 2)

AR-GCN transfers to larger asymmetric problems

Trained on 100 nodes. Tested on larger synthetic TMAT instances.

Task	Reference	500 nodes	1,000 nodes
Asymmetric TSP	LKH-3	**+1.31%**	**+1.64%**
Asymmetric CVRP	HGS	**−1.83%**	**−5.29%**

Lower mean route cost than Hybrid Genetic Search at both ACVRP sizes.

Mean relative gap. Each reference solver was capped at n seconds per instance.

Parameter efficiency and benchmark transfer

0.628M

Learnable parameters

GOAL (2.077M)
Fast T2T (5.334M)

3 epochs

Training per task

200,000 training instances
100 nodes per instance

8.12%

TSPLIB ATSP mean gap

27 benchmark instances
GOAL (25.32% mean gap)

TSPLIB gaps are relative to best-known solutions.

What the solver needs to encode

Re-encode the evolving routing state.

Separate incoming and outgoing edges.

Open question

How can we reduce the cost of full-graph inference on large asymmetric problems?

Project page

github.com/HamJewel/AR-GCN-VRP