TOPological Attention

Edge-conditioned cross-head routing on graphs

Mario Severino*, Riccardo Ali*, Alessio Borgi*, Christopher Irwin*, Pietro Liò

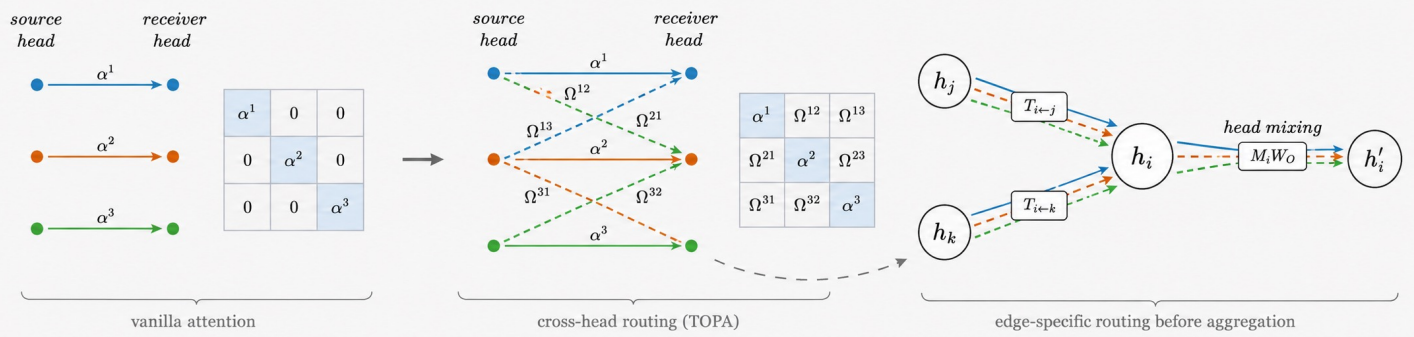
* Equal contribution.

University of Cambridge · Oxford Mathematics · ERLANGEN AI HUB — Mathematical Foundations of Intelligence

Guiding question

What is gained when graph attention moves from diagonal head transport to edge-conditioned matrix-valued transport?

Mechanism: from diagonal attention to edge-specific head transport



Standard attention is diagonal on the head axis

TOPA preserves same-head paths and adds edge-conditioned routes before aggregation

1 Motivation: matrix-valued local transport

Sheaf-inspired perspective
SheafNNs replace scalar edge weights with linear maps between local feature spaces, enabling matrix-valued graph communication.

What is the essential primitive?
Recent evidence leaves unclear whether SheafNN gains come from learned restriction geometry or from the full diffusion construction. At message level, the effective interaction is simply:

$$\rho_{j\to i} = F_{i\to e}^{\top} F_{j\to e} = T_{i\leftarrow j}$$

Our abstraction
We study the directed map itself as local transport, then instantiate it in the natural local space provided by the attention heads.

Full sheaf structure → effective edge map → attention-head transport

2 From sheaf coupling to directed transport

Quiver / graph-copresheaf view
Assign a local space to each node and a directed linear map to each computational arrow:

$$\rho_e : \mathcal{Q}(s(e)) \longrightarrow \mathcal{Q}(t(e))$$

Transport, then aggregate
Incoming states are transformed before they are combined at the receiver:

$$M_i = \sum_{e:t(e)=i} \rho_e X_{s(e)}$$

This isolates the directed edge transport, without assuming a specific factorization, reciprocity, or self-adjoint diffusion of SheafNNs

3 Attention is diagonal directed transport

Attention heads provide the local coordinates
For H heads with C-dimensional values, collect the sender representation as:

$$V_j \in \mathbb{R}^{H\times C}$$

Each edge j→i carries one normalized gate per head:

$$D_{i\leftarrow j} = \mathrm{Diag}\left(\alpha_{i\leftarrow j}^{(1)}, \ldots, \alpha_{i\leftarrow j}^{(H)}\right)$$

Before readout, attention applies this diagonal map and sums:

$$M_i^{\mathrm{att}} = \sum_{j\in\mathcal{N}(i)} D_{i\leftarrow j} V_j$$

Componentwise:

$$\left[D_{i\leftarrow j} V_j\right]^{(r)} = \alpha_{i\leftarrow j}^{(r)} V_j^{(r)}$$

On a given edge, receiver head r only receives source head r. The missing degree of freedom is edge-specific transport between different heads

4 TOPA: from diagonal to cross-head transport

TOPA preserves the original attention and augments it with edge-conditioned off-diagonal transport between heads:

$$T_{i\leftarrow j} = D_{i\leftarrow j}\left(I_H + \lambda\Omega_{i\leftarrow j}\right), \qquad \mathrm{diag}(\Omega_{i\leftarrow j}) = 0$$

Messages are routed before the neighborhood sum:

$$M_i^{\mathrm{TOPA}} = \sum_{j\in\mathcal{N}(i)} T_{i\leftarrow j} V_j$$

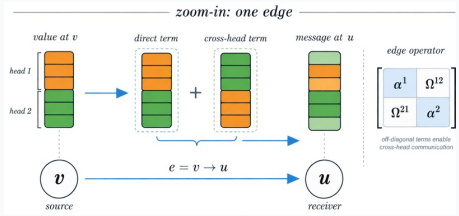
Receiver-head form:

$$\left[M_i^{\mathrm{TOPA}}\right]^{(r)} = \sum_{j\in\mathcal{N}(i)} \alpha_{i\leftarrow j}^{(r)} \left(V_j^{(r)} + \lambda \sum_{s\neq r} \Omega_{i\leftarrow j}^{rs} V_j^{(s)}\right)$$

How to read the factorization
D gates the receiver head; **Ω** chooses which other source heads enter that gate. Setting Ω = 0 recovers the vanilla backbone exactly

Each edge can mix information across heads before contributions from different senders are aggregated

5 One-edge view: what changes locally?



The direct term preserves the same-head path; the cross-head term injects information from other source heads into the receiver-head gate

direct term + cross-head term → routed message

6 Structural checks and backbone integration

A. Diagonal preservation + vanilla recovery

$$\mathrm{diag}(T_{i\leftarrow j}) = \mathrm{diag}(D_{i\leftarrow j})$$

$$\Omega_{i\leftarrow j} = 0 \Longrightarrow T_{i\leftarrow j} = D_{i\leftarrow j}$$

TOPA preserves the original same-head paths and starts from exact functional parity with the matched vanilla backbone

B. Pre-aggregation routing is not ordinary readout mixing

$$\exists A : \sum_j T_{i\leftarrow j} v_j = A \sum_j D_{i\leftarrow j} v_j \iff T_{i\leftarrow j} = A D_{i\leftarrow j}\ \forall j$$

Post-aggregation mixing is equivalent to TOPA only when all incoming edge operators share the same left factor

Because Ω is edge-dependent, different incoming edges can apply different cross-head transformations before their messages are combined

C. Plug-in TOPA enhancement for attention backbones

TOPA augments any multi-head graph-attention layer by modifying only the edge message

$$D_e V_j \longmapsto D_e\left(I_H + \lambda\Omega_e\right)V_j, \qquad \Omega_e = \mathrm{off}\left(\tanh g_\theta(\xi_e)\right)$$

Examples: GATv2 reuses its directed pre-attention context; Graph Transformer reuses its query–key/edge context.
All subsequent backbone operations remain unchanged

Structural message: preserve attention gates, route across heads before aggregation, plug into graph-attention backbones

7 Evaluation protocol: breadth and depth

CLRS — breadth across algorithms
Question: On which graph algorithms does routing help?

Protocol: tuned GATv2 vs TOPA-GATv2; 30 tasks, 20 paired seeds; official task-score differences

STaR — depth of relational composition
Question: Does routing improve extrapolation beyond trained reasoning depths?

Protocol: RCC-8 + Interval Algebra; train b=1–3, k=2–4; test k=2–9; TOPA/full transport vs edge-aware E-GAT; seeds 42–44

b = paths; k = depth. Legacy 15-round shared processor, n=3.
Full transport vs E-GAT is not a nested ablation; identity self-loops are used throughout (b=1*).

Two complementary tests: breadth across algorithms and depth beyond the training regime

8 Results: CLRS task breadth and STaR depth

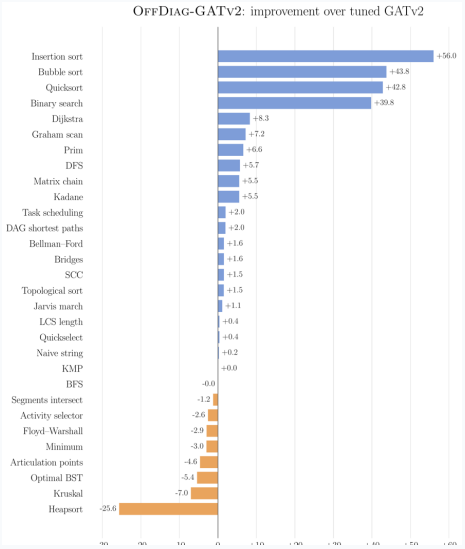
CLRS key signal

21 of 30 tasks improve

mean paired difference: +6.0 points

Mean paired difference across 20 seeds. Strong gains appear on insertion sort, bubble sort, quicksort, and binary search

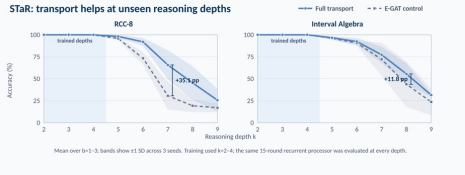
The benefit is selective rather than universal



CLRS: where routing helps
Largest gains concentrate on algorithmic routines that benefit from repeatedly recombining intermediate state: insertion sort (+56.0), bubble sort (+43.8), quicksort (+42.8), and binary search (+39.8)

CLRS: where it interferes
Regressions remain on heapsort (−25.6), Kruskal (−7.0), and Optimal BST (−5.4), showing that more routing is not automatically better

STaR key signal
TOPA/full transport exceeds E-GAT in all 6 domain × seed comparisons. Common-grid gain: +11.35 pp on RCC-8 and +3.38 pp on Interval Algebra. Separation appears at unseen depths k=5–9



Mean over b=1–3; bands show ±1 SD across 3 seeds. Training used k=2–4; the same 15-round recurrent processor was evaluated at every depth.

TOPA is selective, not universally better: CLRS reveals task dependence, while STaR shows its clearest gains beyond trained reasoning depths.