GraphAttnRerank: Attention-Based Re-ranking for Graph Retrieval in RAG

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation with knowledge graphs supports multi-hop reasoning, but subgraphs must be serialized into linear prompts—where each entity block lands interacts with positional bias in large language models. **GraphAttnRerank (GAR)** is a training-free block-to-slot policy: it preserves gold relation-chain order, ranks slots by offline per-slot attention masses $r_\ell(i)$ from calibration probes, and maps blocks to attention-ranked slots in hop order—without altering the retriever or fine-tuning the generator. With retrieval and gold chains held identical across layouts, we compare GAR to baseline layouts, isolating slotting from content.

We verify GAR by testing the following hypotheses (H1–H3): (H1) intermediate chain nodes need attended slots; (H2) attention-ranked placement should beat fixed templates when slot policies match; (H3) chain order should beat similarity order with slot ranking fixed. Experiments on four open-weight LLMs with MuSiQue support these predictions; HotpotQA briefly rechecks. GAR offers a convenient, effective way to exploit positional bias for graph RAG QA: offline r_ℓ steers where chain evidence sits in the prompt, improving accuracy without retraining the retriever or generator.

Keywords: Retrieval-Augmented Generation, knowledge graph, attention re-ranking, multi-hop question answering, structured retrieval

1 Introduction

Retrieval-Augmented Generation (Lewis et al., 2020; Min et al., 2020) combines retrieval with generation for knowledge-intensive NLP; surveys emphasize index design, augmentation depth, and retrieval quality (Gao et al., 2023; Zhao et al., 2022; Peng et al., 2024). Multi-hop question answering stresses composing evidence across reasoning steps (Welbl et al., 2018; Yang et al., 2018). We distinguish *flat* RAG (homogeneous passage lists) from *structured* RAG, where retrieval exposes entities, relations, or subgraphs that must be linearized into the prompt (Edge et al., 2024; Chen et al., 2025b; Han et al., 2025). Unlike sequences of chunks, a subgraph admits many display orders; a hop-aligned relation chain fixes logical precedence before blocks are mapped to slots $1, \ldots, K$. Display order need not equal hop index—the design choice we study is *which slot holds which chain block*.

Two factors affect downstream accuracy: retrieval content and *prompt layout*. Decoder LLMs exhibit positional bias: attention mass from answer-related positions is uneven over input positions (Liu et al., 2023; Cuconasu et al., 2025). Prior work improves lists by reranking (Liu et al., 2024; Yi et al., 2025) or permutations (Tang et al., 2024); calibration can mitigate "lost in the middle" effects (Hsieh et al., 2024). After serializing a subgraph along a relation chain, one must still assign hop-aligned blocks to slots $1, \ldots, K$; we study *block-to-slot assignment* while holding the multiset of entity-grounded blocks fixed. GraphAttnRerank (GAR) uses offline $r_\ell(i)$ at a chosen layer ℓ to rank slots and assign blocks, without fine-tuning the generator.

GAR does not train new retrievers, graph extractors, or decoders; it optimizes layout once chains are identified, in line with settings where retrieval and generator weights are fixed. Calibration produces a length-K vector of slot statistics per checkpoint and may be rerun when the generator is replaced.

We evaluate GAR under identical items and chains across layouts so differences reflect slotting. Unlike permutation sampling methods (Tang et al., 2024), GAR uses one deterministic layout per item for analysis and efficient decoding.

Contributions. (1) GAR: rank slots by offline r_ℓ, assign chain-faithful blocks to high-r_ℓ slots

(Figure 2), without training the generator. (2) A protocol tying calibration probes to graph extraction outputs so σ and layouts are reproducible. (3) Hypothesis-targeted evaluation with controlled retrieval (Tables 4–7; HotpotQA transfer in Table 8), isolating placement under matched items and gold chains across layouts.

Roadmap. Background and hypotheses (§2); related work (§3); method (§4); experiments (§5); discussion and positioning (§§6–7); conclusion (§8). Appendices A–F give prompts, calibration figures, statistics, and robustness tables.

2 Background and Formalization

2.1 Positional attention statistics

Let layer ℓ yield attention matrices $A^{(\ell)} \in [0,1]^{L\times L}$ over prompt length L (rows normalized). A calibration prompt has K blocks $i \in \{1,\dots,K\}$ occupying token index sets $\mathcal{I}_i$. On calibration instance c, define mean incoming attention to block i:

$$\tilde{r}_\ell^{(c)}(i) = \frac{1}{L}\sum_{j=1}^{L}\frac{1}{|\mathcal{I}_i|}\sum_{t\in\mathcal{I}_i} A_c^{(\ell)}(j,t)\,. \qquad (1)$$

Aggregate over calibration set $\mathcal{C}$:

$$r_\ell(i) = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \tilde{r}_\ell^{(c)}(i)\,. \qquad (2)$$

Higher $r_\ell(i)$ indicates a slot where evidence receives more incoming attention mass *on average*, motivating placement of critical blocks there.

2.2 Positional curves

Figure 1 plots last-layer $r_\ell(i)$ used for GAR on four checkpoints across $K \in \{5, 10, 15\}$ (Llama-3.1-8B-Instruct ℓ=31; Qwen3-8B ℓ=35; Gemma-3-4B-IT layer 30 global attention; Phi-4-mini last full-attention layer). Appendix B shows the full layer-wise sweeps used to justify these layer choices. Curve shape varies by architecture and K.

Table 1 reports normalized positional entropy $H_\ell/\log_2 K$ on the same probes; lower values indicate sharper concentration of attention mass over slots.

Model	K=5	K=10	K=15	Overall
Llama-3.1-8B	0.972	0.962	0.958	0.964
Qwen3-8B	0.959	0.935	0.713	0.869
Gemma-3-4B-IT	0.847	0.613	0.511	0.657
Phi-4-mini	0.694	0.628	0.825	0.716

Table 1: Normalized entropy $H_\ell/\log_2 K$ of last-layer slot attention (same probes as Figure 1).

2.3 GAR, baselines, and hypotheses

GraphAttnRerank (GAR) ranks relation chains (e.g. by chain–query similarity), sorts slots by $r_\ell(i)$ descending to obtain σ, then assigns each chain's entity-grounded nodes to the next available highest-r_ℓ slots while preserving intra-chain order. Local follows a GraphRAG-style fixed markdown layout after grounding; Desc/Asc place similarity-ranked blocks at slots $1..K$ in list order without σ alignment; AAS lists blocks by descending query–block similarity, then applies the same σ as GAR—isolating chain order vs. similarity order under identical slot policy.

H1 (intermediate nodes): forcing bridge entities into lowest-r_ℓ slots (*Middle-at-lowest*) should hurt continuity vs. Local. H2 (attention-aligned placement): AAS should improve over Local when σ reallocates mass to informative blocks. H3 (chain order): GAR should beat AAS when relation-chain order carries reasoning signal beyond similarity sorting.

3 Related Work

3.1 RAG, graphs, and serialization

Dense retrieval and generation (Lewis et al., 2020) and graph-augmented pipelines (Edge et al., 2024; Li et al., 2026; Peng et al., 2024) foreground *what* enters context. Once entities and relations are verbalized, serialization order remains underspecified; we hold retrieval fixed and vary only slotting.

3.2 Multi-hop QA and path-centric retrieval

Multi-hop benchmarks (Yang et al., 2018; Welbl et al., 2018) stress evidence composition. Path-centric methods such as PathRAG (Chen et al., 2025a) retrieve relational paths; ReAttn-style modules (Tian et al., 2026) re-weight retrieved units. These layers decide candidates and scores; they do not prescribe which prompt slot hosts which hop-aligned block given a fixed multiset.

3.3 Position, calibration, and layout control

Long-context utility is position-sensitive (Liu et al., 2023; Gupte et al., 2025); RAG systems exhibit analogous positional effects on retrieved lists (Cuconasu et al., 2025). Mitigations include training-time positional calibration (Hsieh et al., 2024). Permutation-based decoding (Tang et al., 2024) explores orderings stochastically; GAR fixes a single

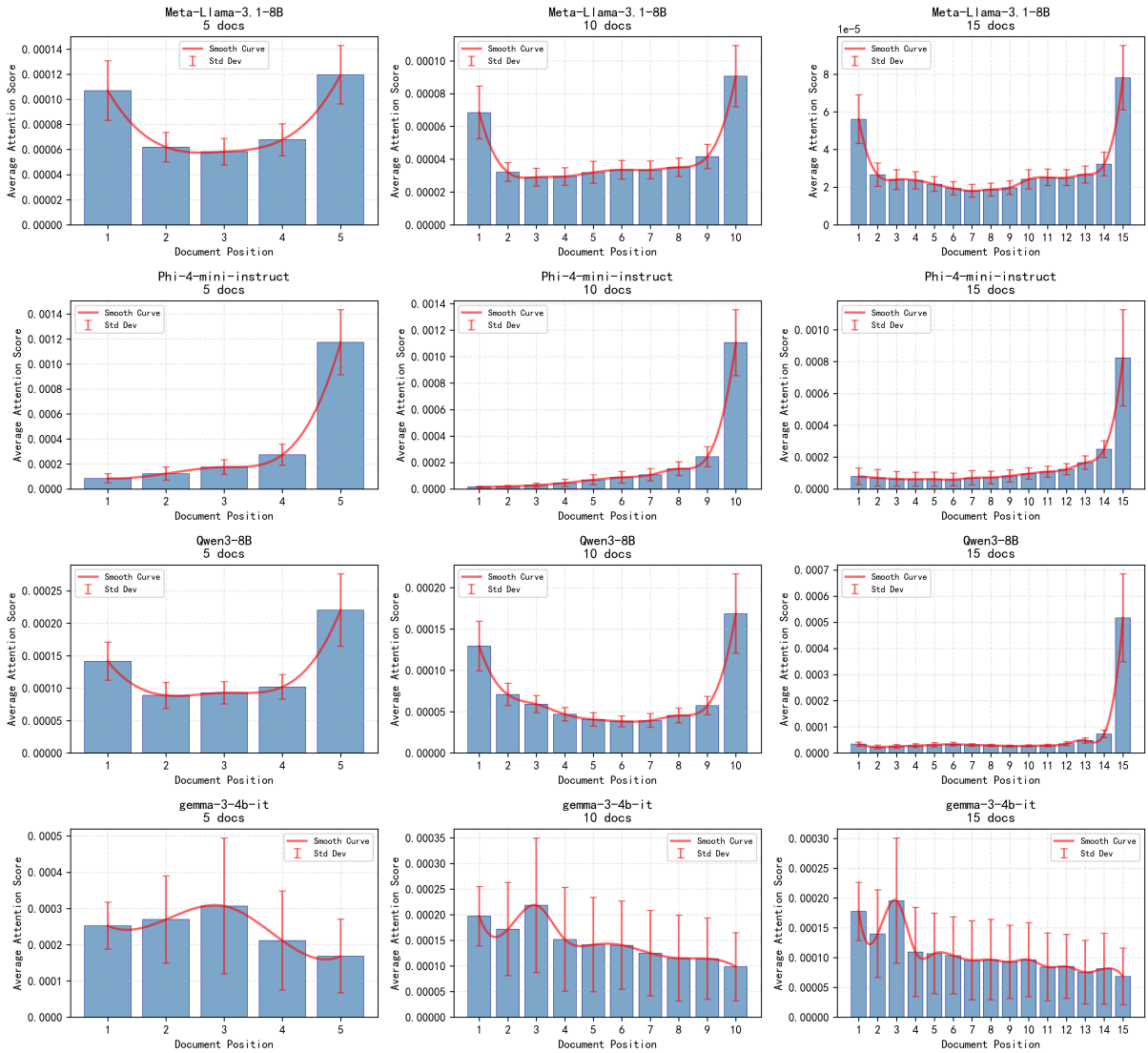

Figure 1: Last-layer per-slot mean incoming attention $r_\ell(i)$ vs. slot index i for $K \in \{5, 10, 15\}$.

deterministic σ from r_ℓ matched to the generator family.

3.4 List reranking vs. block-to-slot placement

Attention-based rerankers (Liu et al., 2024; Chen et al., 2026) and AttnRank (Yi et al., 2025) reorder *lists* of passages or documents. GAR fixes the multiset of chain blocks and solves *placement*: which slot index receives which block under a chosen listing policy—orthogonal to scoring which items to retrieve.

Calibration, reranking, and layout. Mitigations for positional bias span *training-time* updates that target long-context attention utilization (Hsieh et al., 2024); *attention-informed reranking*, which scores or permutes candidates (often with attention-derived signals) without prescribing block-to-slot maps over a fixed chain multiset (Liu et al., 2024; Chen et al., 2026; Yi et al., 2025); and *inference-only layout*, where offline statistics guide placement while generator weights stay frozen. GAR falls in the third class: $r_\ell(i)$ ranks slots and maps a fixed multiset of chain-grounded blocks—distinct from reranking *which* units enter context and from calibration that edits model parameters.

4 Method

Calibration. We estimate $r_\ell(i)$ with $|\mathcal{C}|$=500 synthetic probes per model and K, each probe instantiating K neutral document blocks (100–500 tokens), independent of evaluation questions. Chain-query similarity and query–block similarity for AAS/Desc/Asc use Qwen3-Embedding-4B cosine similarity (shared API settings).

Inference pipeline. Figure 2 mirrors the imple-

Algorithm 1 GAR: attention-aligned block-to-slot assignment

Input: K prompt slots; $r_\ell(i)$ for $i{=}1, \ldots, K$; query q; retrieved relation chains.
Step 1. Rank chains by chain-query similarity; keep top chains covering the evaluation protocol.
Step 2. Sort all K slots by $r_\ell(i)$ descending (ties broken deterministically).
Step 3. For each chain L_j in chain-ranking order: let k_j be its entity-grounded node count; take the k_j highest-r_ℓ slots still unused; place nodes in chain order into those slots left-to-right.
Output: slot assignment for generation.

mentation: offline calibration → chain ranking → slot ranking σ → GAR assignment → generation. Baselines share retrieval, chain identification, and decoding (temperature 0, fixed seed).

Integration and cost. GAR inserts after retrieval and chain ID; it does not alter retriever or generator weights. Sorting slots costs $O(K \log K)$; similarity features for AAS/Desc/Asc add $O(K)$ embedding calls. Memory stores only length-K vectors $r_\ell(i)$.

Notation. Let σ be the slot permutation induced by sorting $\{1, \ldots, K\}$ by $r_\ell(i)$ descending. GAR processes chains $(L_1, \ldots)$ in exogenous chain-ranking order; for each L_j it consumes the next available highest-r_ℓ slots. AAS instead forms a *single* list of blocks ordered by query–node cosine similarity, walks that list, and fills σ sequentially—breaking explicit chain adjacency unless similarity happens to preserve it.

Entity-grounded blocks. Each hop contributes one block with entity-linked surface text; distractors occupy remaining slots under each condition's policy. Graph extraction and chain supervision follow the preprocessing pipeline released with this submission so that all layouts consume identical tokens modulo permutation.

5 Experiments

5.1 Setup

Data. Primary evaluation uses MuSiQue-derived graph QA: LLM-based graph extraction yields hop-aligned chains and entity-grounded spans; items are stratified by gold hop count (2/3/4) so shallow and deep reasoning are not averaged away. HotpotQA uses the same entity-grounded ordering pipeline for out-of-domain robustness.

Procedure. Each item fixes retrieval and the gold chain; we vary one layout, build the prompt, and query the generator (no test-time fine-tuning).

Variant	Description
Local	GraphRAG-style fixed layout after grounding (no σ).
Desc / Asc	Similarity-ordered blocks at slots $1..K$.
AAS	Similarity block order, then σ from r_ℓ.
GAR	Chain order, then σ from r_ℓ.
Mid-low	H1 stress test: force intermediates to lowest-r_ℓ slots.

Table 2: Layout conditions (shared retrieval and chains).

Item	Setting
Models	Llama-3.1-8B-Instruct, Qwen3-8B, Gemma-3-4B-IT, Phi-4-mini (OpenAI-compatible API).
K / hops	$K \in \{5, 10, 15\}$; evaluation across 2/3/4-hop items.
Calibration	500 probes/model/K; layer ℓ per model as in §2.
Primary tables	T1–T4 (§§5.2–5.5); HotpotQA Table 8.

Table 3: Experimental protocol summary.

Conditions appear in Table 2. Metrics use an LLM judge (Zheng et al., 2023) (greedy Yes/No on the first line; retries on parse failure). Tables 4–7 report aggregate accuracies; win-rate rows summarize per-hop contrasts (Appendix C).

5.2 Main comparison (T1)

Table 4 compares Local, similarity orders, and GAR. GAR improves over Local on average (+1.87 pp) with an 11/12 win rate across model–K cells, and improves over Desc/Asc on average (+0.60 / +0.82 pp; 9/12 each), indicating attention-informed slotting helps even when block lists are similarity-sorted.

5.3 H1: Intermediate nodes (T2)

Table 5 contrasts *Middle-at-lowest* with Local. Δ is negative in 5/12 cells and positive in 7/12, with larger swings on Qwen3 and Phi at $K \in \{10, 15\}$—consistent with bridge entities interacting strongly with where attention mass concentrates.

5.4 H2: Attention-aligned placement (T3)

Table 6 reports AAS margins vs. Local/Desc/Asc and Mid-low. Average AAS−Local is +0.27 pp (6/12 positive cells); AAS improves over Mid-low on average (+0.22 pp; 7/12), supporting H2 that σ reallocates attention toward useful blocks even when chain order is not yet enforced.

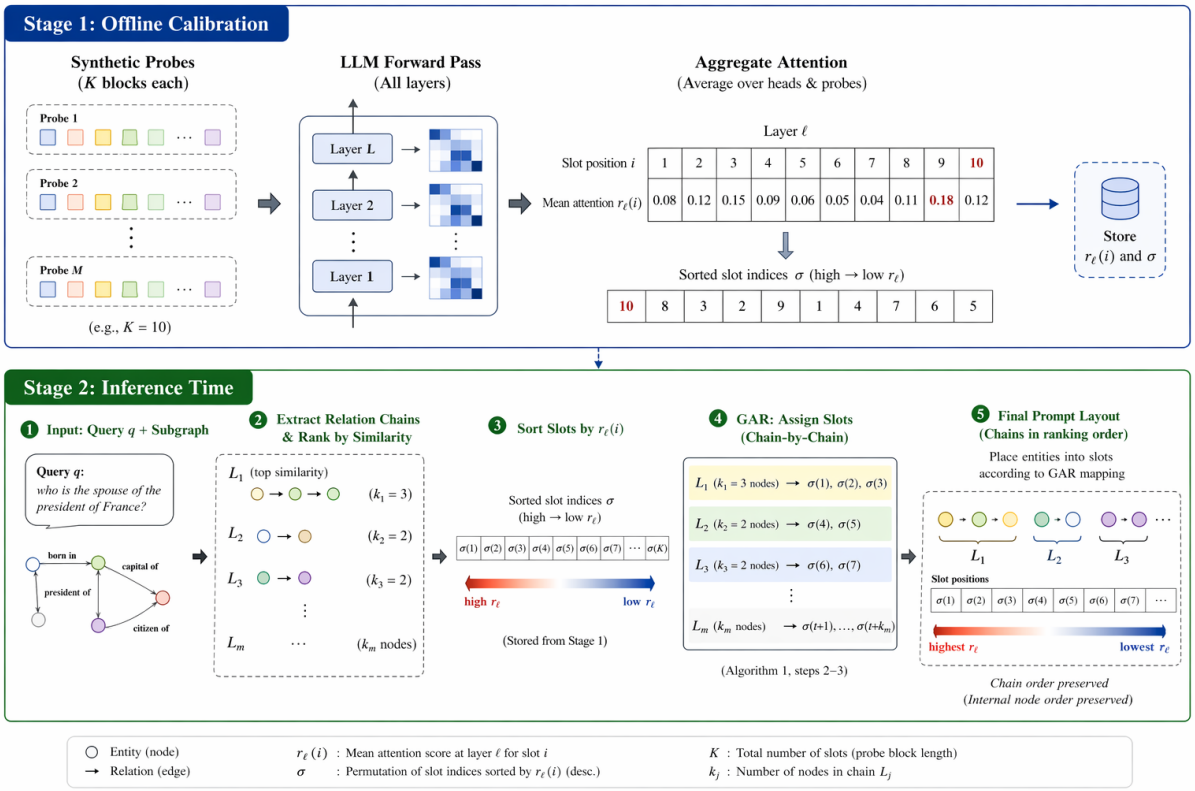

Figure 2: GAR pipeline: calibration yields $r_\ell(i)$ and attention-ranked slots; inference ranks chains, assigns nodes chain-by-chain to the highest remaining attention-ranked slots, then queries the LLM.

Model	K	Local	Desc	Asc	GAR
Llama	5	0.908	0.941	0.937	0.946
Llama	10	0.892	0.878	0.878	0.888
Llama	15	0.887	0.896	0.900	0.916
Qwen3	5	0.941	0.951	0.952	0.950
Qwen3	10	0.921	0.920	0.918	0.927
Qwen3	15	0.925	0.925	0.910	0.930
Gemma	5	0.904	0.916	0.921	0.920
Gemma	10	0.849	0.904	0.881	0.904
Gemma	15	0.850	0.892	0.887	0.889
Phi	5	0.929	0.930	0.933	0.931
Phi	10	0.894	0.898	0.903	0.905
Phi	15	0.887	0.888	0.893	0.904
Avg		0.899	0.912	0.909	0.918

Table 4: **T1:** Overall accuracy of four layout conditions (mean-of-hop strata per cell). *Avg*: unweighted mean over 12 model–K cells.

Model	K	Local	Mid-low	Δ(pp)
Llama	5	0.908	0.898	−1.01
Llama	10	0.892	0.902	+1.02
Llama	15	0.887	0.897	+0.99
Qwen3	5	0.941	0.944	+0.30
Qwen3	10	0.921	0.910	−1.14
Qwen3	15	0.925	0.907	−1.75
Gemma	5	0.904	0.920	+1.62
Gemma	10	0.849	0.866	+1.72
Gemma	15	0.850	0.863	+1.34
Phi	5	0.929	0.919	−1.02
Phi	10	0.894	0.883	−1.19
Phi	15	0.887	0.884	−0.31

Table 5: **T2:** H1—middle-at-lowest vs. Local (overall accuracy; Δ in pp).

5.5 H3: Chain order vs. similarity order (T4)

Table 7 holds σ fixed and swaps only block listing policy: GAR (chain) vs. AAS (similarity). Per-hop contrasts yield 28/32 wins for GAR vs. AAS (+1.60 pp mean), supporting H3.

5.6 HotpotQA robustness

Table 8 summarizes the same layout protocol on HotpotQA (count-weighted overall accuracy). GAR most often improves over Local; gains remain model- and baseline-dependent, consistent with cross-domain transfer limits.

5.7 Findings overview

H1 is mixed by model and K (positional curves differ); **H2** favors attention-aligned placement on aggregate; **H3** favors chain order over similarity order under shared σ. Headline margins appear in Tables 4–7; Appendix C gives bootstrap CIs and full per-hop grids.

Model	K	A−L	A−D	A−A	A−M
Llama	5	−0.26	−0.26	+0.40	+0.76
Llama	10	+1.12	+0.40	+0.47	+0.10
Llama	15	+0.92	+0.17	−0.06	−0.07
Qwen3	5	+0.76	+0.66	+0.31	+0.46
Qwen3	10	−0.96	−0.54	−0.33	+0.18
Qwen3	15	−1.89	−0.34	−0.51	−0.15
Gemma	5	+0.51	+0.01	+0.26	−1.11
Gemma	10	+3.53	+0.41	+1.64	+1.81
Gemma	15	+2.54	+0.42	+0.35	+1.20
Phi	5	−1.12	−1.07	−0.26	−0.11
Phi	10	−1.05	−0.11	+0.11	+0.14
Phi	15	−0.92	−0.16	+0.10	−0.60
Avg		+0.27	−0.03	+0.21	+0.22

Table 6: **T3:** H2—AAS Δ vs. Local (L), Desc (D), Asc (A), Mid-low (M), pp.

Model	K	GAR	AAS	Loc	Dsc	Asc
Llama	5	0.946	0.906	0.931	0.941	0.937
Llama	10	0.888	0.903	0.883	0.878	0.878
Llama	15	0.916	0.896	0.894	0.896	0.900
Qwen3	5	0.950	0.948	0.945	0.951	0.952
Qwen3	10	0.927	0.912	0.919	0.920	0.918
Qwen3	15	0.930	0.906	0.928	0.925	0.911
Gemma	5	0.920	0.909	0.909	0.916	0.921
Gemma	10	0.904	0.885	0.859	0.904	0.881
Gemma	15	0.889	0.875	0.859	0.892	0.887
Phi	5	0.931	0.918	0.939	0.930	0.933
Phi	10	0.905	0.884	0.897	0.898	0.903
Phi	15	0.904	0.878	0.894	0.888	0.893
Avg		0.918	0.902	0.905	0.912	0.909
WR			28/32	26/32	25/32	23/32

Table 7: **T4:** H3—overall accuracy; WR = per-hop win counts vs. column (32 strata).

5.8 Implementation notes

Decoding, calibration, sharded `summary.json` artifacts, and the Excel merge pipeline are specified in Appendix A; all layouts share tokenizer settings and fixed boundary markers per checkpoint.

5.9 Statistical reporting conventions

Appendix C states how MuSiQue tables pool hop strata (unweighted means per script) versus HotpotQA count-weighting, documents paired-bootstrap methodology, and reminds readers that baseline columns are not interchangeable across tables—only within-table contrasts are directly comparable.

6 Discussion

6.1 When does GAR help?

GAR tends to pay off when the ingredients assumed by calibration hold at deployment time: the prompt uses the same slot count K as calibration, the curve $r_\ell(\cdot)$ assigns differentiated mass across slots so σ is informative, and calibration and inference share the same generator family so slot statistics remain comparable. Normalized positional entropy correlates with marginal gains: Gemma shows the lowest entropy among the four models and larger lifts vs. Local in several regimes, consistent with peaked r_ℓ curves (Figure 1). Empirical validation at the target K is advisable after checkpoint changes because gains vary non-monotonically with K for some models (§6.4).

6.2 Hop depth and model interaction

At 4-hop depth, lift vs. Local varies by architecture (e.g. Gemma +6.04 pp vs. Llama +1.17 pp in aggregated hop analyses), underscoring that deeper chains stress slot competition—calibration coverage should include representative chain lengths.

6.3 Judge variance and effect sizes

Binary judge outcomes near 90%+ accuracy compress measurable Δ; small pp shifts may lie inside bootstrap CIs unless sample sizes grow (Zheng et al., 2023; Tan et al., 2024). Appendix C summarizes resampling; we mitigate via stratification and transparent tables; human evaluation remains complementary. When the judge model is updated, comparable stratification and reporting should be preserved.

6.4 Model- and K-specific behavior

Even under identical σ, Meta-Llama-3.1-8B-Instruct shows non-monotonic K dependence (strong wins at K=5 and 15 but not at 10 in several contrasts), whereas Qwen3 and Phi behave more smoothly. We hypothesize interactions among rotary encodings, absolute positions, and chain length; verifying this requires controlled interventions beyond this work's scope.

7 Relation to list reranking and training-time mitigation

§3 contrasts list-level reranking with block-to-slot placement. Operationally: rerankers rescore *which* passages enter the list (Liu et al., 2024); GAR never rescores retrieval but permutes a fixed multiset of chain-aligned blocks across slots using r_ℓ. Training-time positional calibration (Hsieh et al., 2024) edits model weights; GAR consumes offline statistics only ($O(K \log K)$ sorting at inference). GAR and AAS jointly isolate chain vs. similarity

Model	K	GAR	Local	Desc	Asc	Δ_L	Δ_D	Δ_A
Gemma	5	79.37	77.76	80.24	80.36	+1.61	−0.87	−0.99
Gemma	10	81.91	78.81	81.72	80.05	+3.10	+0.19	+1.86
Gemma	15	81.47	79.37	80.42	80.30	+2.11	+1.05	+1.18
Llama	5	80.24	79.55	79.49	80.24	+0.69	+0.75	0.00
Llama	10	82.34	80.85	81.91	81.97	+1.49	+0.43	+0.37
Llama	15	82.84	82.09	81.72	82.28	+0.74	+1.12	+0.56
Phi	5	82.78	81.91	81.60	83.52	+0.87	+1.18	−0.74
Phi	10	82.84	82.46	81.54	82.71	+0.38	+1.30	+0.13
Phi	15	82.40	81.72	81.10	81.41	+0.68	+1.30	+0.99
Qwen3	5	84.76	84.57	84.82	84.95	+0.19	−0.06	−0.19
Qwen3	10	85.87	86.55	85.63	86.25	−0.68	+0.24	−0.38
Qwen3	15	86.06	85.75	85.07	86.68	+0.31	+0.99	−0.62

Table 8: HotpotQA overall accuracy (%) and GAR deltas vs. Local / Desc / Asc (percentage points).

listing under a fixed σ, which sufficed to replay layouts in our experiments.

8 Conclusion

We presented GraphAttnRerank: offline $r_\ell(i)$ induce slot order σ; GAR maps relation-chain blocks onto high-attention slots without fine-tuning the generator. Results support the role of intermediates, σ, and chain ordering under controlled retrieval (with partial HotpotQA transfer).

Future work includes richer calibration when K or checkpoints change, and coupling with path-centric retrieval (Chen et al., 2025a) under fixed inference budgets.

9 Limitations

- **White-box access:** GAR requires attention tensors for calibration; closed APIs without attention exports are unsupported without aligned surrogates.
- **Chain correctness:** Misidentified chains cannot be repaired by slotting alone.
- **Cost:** Hundreds of forward passes amortize across deployment but remain non-trivial at large scale.
- **Judge bias:** LLM judges inherit known biases; reported Δ should be read with CI awareness at high baselines.
- **English-centric probes:** Calibration templates are English-neutral; multilingual deployments should rebuild $\mathcal{C}$ in target locales.
- **Concurrent structure edits:** If entity linking or coreference pipelines drift between calibration and serving, slot masses may no longer reflect production prompts.

Ethical considerations

Our study uses public benchmarks and synthetic probes only; no user-generated logs or private documents are processed. Misuse risk under typical QA settings is limited, but hallucination rates should still be monitored independently of layout modules. We disclose LLM judge usage and include prompts for replication.

Responsible NLP checklist. Completed in the OpenReview submission form (per ARR).

Reproducibility. Pipeline commands and prompt templates appear in Appendix A.

References

Boci Chen and 1 others. 2025a. Pathrag: Pruning graph-based retrieval augmented generation with relational paths. *arXiv preprint arXiv:2502.14902*.

Haonan Chen and 1 others. 2026. Where relevance emerges: A layer-wise study of internal attention for zero-shot re-ranking. *arXiv preprint arXiv:2602.22591*.

Ruicheng Chen and 1 others. 2025b. Retrieval-augmented generation with knowledge graphs: A survey. *OpenReview (CSUC 2025)*. Available: https://openreview.net/forum?id=ZikTuGY28C.

Florin Cuconasu and 1 others. 2025. Do rag systems really suffer from positional bias? In *EMNLP*. ArXiv:2505.15561.

Darren Edge and 1 others. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Yunfan Gao, Yun Xiong, and 1 others. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Mrunal Gupte and 1 others. 2025. What works for 'lost-in-the-middle' in llms? a study on gm-extract and mitigations. *arXiv preprint arXiv:2511.13900*.

Hanyu Han and 1 others. 2025. Retrieval-augmented generation with graphs (graphrag). *arXiv preprint arXiv:2501.00309*.

Cheng-Yu Hsieh and 1 others. 2024. Found in the middle: Calibrating positional attention bias improves long context utilization. In *ACL Findings*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yuchen Li and 1 others. 2026. Deep graphrag: A balanced approach to hierarchical retrieval and adaptive integration. *arXiv preprint arXiv:2601.11144*.

Nelson F. Liu and 1 others. 2023. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*.

Zhengyang Liu and 1 others. 2024. Attention in large language models yields efficient zero-shot re-rankers. *arXiv preprint arXiv:2410.02642*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2020. Realm: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning (ICML)*.

Boci Peng and 1 others. 2024. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*.

Shiyang Tan and 1 others. 2024. Judgebench: A benchmark for evaluating llm-based judges. *arXiv preprint arXiv:2410.12784*.

Zhengyang Tang and 1 others. 2024. Found in the middle: Permutation self-consistency improves listwise ranking in large language models. In *NAACL*. ArXiv:2310.07712.

Yu Tian and 1 others. 2026. Reattn: Improving attention-based re-ranking via attention re-weighting. *arXiv preprint arXiv:2602.19969*.

Johannes Welbl and 1 others. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics (TACL)*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Zeqian Yi and 1 others. 2025. Attention basin: Why contextual position matters in large language models. *arXiv preprint arXiv:2508.05128*. Introduces AttnRank.

Wayne Xin Zhao and 1 others. 2022. A survey on retrieval-augmented text generation. *ACM Computing Surveys*.

Lianmin Zheng and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Appendix A Reproducibility Notes

A.1 Data and evaluation setup

Primary corpus (§ 5.1): MuSiQue-derived graph QA (answer-bearing multi-hop QA with extracted entities, relations, and a gold-aligned relation chain). The test split contains 1,000 2-hop, 1,000 3-hop, and 644 4-hop items (total 2,644).

Generation parameters (all generators): temperature = 0.0, max_new_tokens = 350, seed = 42. LLM judge (Qwen3-30B-A3B-Instruct-2507): temperature = 0.0. Judge prompt follows a binary Yes/No format requiring the first line to start with "Yes" or "No" (non-conforming replies are retried up to 3 times, then scored as incorrect).

Answer extraction: ground-truth and predicted answers are extracted from model output via `<answer>...</answer>` tags; fallback patterns include `answer:\s*(.+?)(?:\n|$)`, `answer is:\s*(.+?)(?:\n|$)`, and `the answer is:\s*(.+?)(?:\n|$)`. If no tag is found, the raw output is used.

Embedding model: Qwen3-Embedding-4B (cosine similarity; same embedding setup as for retrieval in the main experiments).

Retrieval mode: entity-based (entity-grounded blocks), not path-based. H3 contrasts two block listings before rank–rank (names and shorthand in § 5.5, Table 7). Results in the main text (Tables 4–7, § 5.2–5.5) use the GAR configuration in § 5.5 unless otherwise noted.

A.2 Attention calibration

Attention statistics are stored as per-layer, per-block-index mean incoming attention over calibration probes with K document blocks. Each probe renders K entity-grounded blocks in a fixed template order (random or fixed layout) and averages the LLM's last-token attention vector over all positions within each block.

Layer choice: last full-attention layer of the target model. For Gemma-3-4B-IT, the physical last layer (layer 29 in the architecture) is sliding-window only; GAR uses layer 30 (the last global-attention layer) for this model. All other models use layer -1 (last in the stack).

A.3 Slot variants and hypotheses

Four slot variants are tested (§ 5.1): Fixed local layout (GraphRAG local–style fixed template order), Similarity descending (blocks assigned to slots by descending query–block similarity), Middle-at-lowest (chain intermediates forced to lowest-attention slots; stress test for H1), and GAR (Algorithm 1: relation-chain entity branch, rank–rank to attention-ranked slots).

Hypotheses H1–H3 are validated through contrasts between these variants as specified in the table in § 5.1. H3 (§ 5.5, Appendix C) is tested by contrasting the two H3 block listings in § 5.5 under the same σ policy.

A.4 Robustness and reporting

Robustness: Table 7 (§ 5.5) reports GAR vs baseline Δ at K = 5, 10, 15 (three retrieval budget settings). All reported values are in percentage-point accuracy difference ($\times 100$). Statistical note: ± 1 pp can fall within noise range for this evaluation size; bootstrap confidence intervals are reported in Appendix C.

Figures: Figure 1 = last-layer $r_\ell(i)$ curves (four models); Appendix B (Figures B1–B4) = all transformer layers ℓ for the same probes, grouped by model with $K{=}5, 10, 15$ panels, enabling comparison of early vs deep positional shapes and verification of the last-layer choice.

Hyperparameters not swept: the H3 block-listing contrast (§ 5.5) is evaluated but not tuned; probe count is fixed at 100–500 tokens per block; no hyperparameter optimization was performed on the generator side beyond temperature = 0.0 and seed = 42.

A.5 Exact prompts

Below are the precise prompts used at each stage of the procedure. Placeholders (curly braces) are replaced at runtime with per-sample data.

A.1 Calibration probe prompt (for computing $r_\ell(i)$) During calibration, documents are rendered in a fixed order; the forward pass yields mean last-token attention per block index i.

Please read the following documents carefully and answer the question.
Documents:
Document [1] {doc_1}
Document [2] {doc_2}
...
Document [K] {doc_K}

Question: {question}

After tokenization and model forward pass, per-head attention tensors are averaged over heads, then averaged over all tokens belonging to each document block. The result is $r_\ell(i)$ for layer ℓ and block position i.

A.2 Entity block text (per block) Each entity-grounded block is rendered as follows (used in both the calibration probes and the evaluation prompts):

Entity: {entity_name}
Description: {entity_description}
Relations:
{entity} -> {target}: {relation_description}
...

Single-hop entities (no outgoing relations) render without the Relations section.

A.3 Evaluation prompt (generation) The prompt is shown below as a template for each variant.

Format A — non-local layout variants (descending, ascending, GAR, middle-at-lowest, random):
Answer question based on the following information.

[Node 1]
Entity: {entity_1}
Description: {desc_1}
Relations:
{entity_1} -> {target}: {rel_desc}
...

[Node 2]
Entity: {entity_2}
...
[Node K]
...

Question:{question}

Format B — GraphRAG local layout baseline:
Answer question based on the following information.

Entity	Description
———	————-
{entity_1}	{desc_1}
{entity_2}	{desc_2}
...
| {entity_K} | {desc_K} |

Relationships

{entity_1} -> {target}: {rel_desc}
{entity_2} -> {target}: {rel_desc}
...

Question:{question}

The generator is instructed to output <answer>...</answer> tags wrapping the answer.

A.4 LLM judge prompt The judge receives the question, reference answer, and predicted answer; replies must start with "Yes" or "No" on the first line.

You are a fair evaluator for question answering.
Decide whether the Predicted Answer is correct with respect to the Question, using the Reference Answer as the gold standard.
Guidelines: - The Predicted Answer may be long or include reasoning; judge whether it asserts the same core fact as the Reference Answer (same entity, number, date, or clearly equivalent wording). - If the reference is a short name or phrase, answer Yes when the prediction clearly gives that answer, even if it appears inside a longer explanation. - Extra reasoning or context alone is not a reason to answer No if the factual answer matches. - Answer No only if the prediction contradicts the reference, gives a different main answer, or omits the correct answer.
Question: {question} Reference Answer: {reference} Predicted Answer: {prediction}
Reply with exactly one word on the first line: Yes or No.

Model	Layer ℓ	K=5	K=10	K=15	Overall
Meta-Llama-3.1-8B-Instruct	31	97.2%	96.2%	95.8%	96.4%
Qwen3-8B	35	95.9%	93.5%	71.3%	86.9%
Gemma-3-4B-IT	30	84.7%	61.3%	51.1%	65.7%
Phi-4-mini-instruct	31	69.4%	62.8%	82.5%	71.6%

Non-conforming replies (first line does not start with "Yes" or "No") trigger up to 3 retries. Persistent failure is treated as incorrect (0.0).

A.5 Graph extraction prompts (data construction, § 5.1) Used when constructing the graph-QA corpus from MuSiQue source data.

Paragraph-level entity/relation extraction:

Extract entities and relations from the following paragraph. Output pure JSON.

[Input] Title: {title} Text: {text}

[Rules] - Extract named entities (people, organizations, places, works, etc.) with brief descriptions. - Extract relations between entities: source, target, and a 1-2 sentence description of the relation. - Relations can be within the paragraph or to entities mentioned (e.g. "X is the spouse of Y"). - Output format: {{"entities": [{{"name": "X," "description": "...”}}], "relations": [{{"source": "X," "target": "Y," "description": "...”}}]}}

[Output] Pure JSON only, no markdown.

Cross-paragraph graph linking (with answer chain):

Given the answer chain and extracted entities from multiple paragraphs, output a unified graph.

[Answer chain] (from question decomposition, must be preserved): {chain_json}

[Extracted entities and relations per paragraph] {extractions_json}

[Rules] - Merge entities that refer to the same thing. - Add cross-paragraph relations: when entity A in paragraph 1 connects to entity B in paragraph 2. - Preserve the answer chain entities with flag 0,1,2,... (start->mid->...->answer). - CRITICAL: The answer path must have exactly {num_answer_entities} entities. Do NOT add extra intermediate entities. - Add other entities as flag=-1 for distractors. - Each relation needs source, target, description (>=20 chars).

[Output] Pure JSON: {{ "entity_blocks": [{{ "entity": "X," "description": "...," "flag": 0, "relations": [{{ "source": "X," "target": "Y," "description": "...” }}] }}], "distractor_paths": [{{ "source": "A," "target": "B," "description": "...” }}] }} entity_blocks: answer path entities (flag>=0) first, then distractor entities (flag=-1). Exactly 15 distractor_paths, each 2-4 edges.

Appendix B Positional Curves by Layer (Full Calibration Geometry)

B.1 B.0 Positional entropy

Following Yi et al. [31], we build synthetic multi-block prompts (probes), run the target LLM forward, and aggregate positional attention into per-slot curves $r_\ell(i)$ as in § 2. We sweep block count K and layer ℓ so that which layer's statistics to use is an empirical choice, not a default to the first transformer block. The main text (Figure 1) shows only the last-layer curves used for GAR on four models; this appendix gives the full layer-by-layer positional plots (and the same K sweeps), so the complete calibration geometry is available without crowding the body.

Our findings differ from AttnRank [31] in two respects. First, AttnRank reports a U-shaped positional curve in the first layer of Llama; we observe U-shaped (or otherwise non-uniform) curves at the last full-attention layer, which is the layer we use for slot ranking. Second, the shape of the positional attention distribution varies across model families: Gemma exhibits a sharply peaked curve while Llama and Qwen3 are substantially flatter, indicating that the attention geometry is architecture-dependent rather than universal.

Positional entropy. To quantify the sharpness of the positional attention curve, we compute the normalized positional entropy of the probability distribution $p_i = \hat{r}_\ell(i)$:

$$H_\ell \;=\; -\sum_{i=1}^{K} \hat{r}_\ell(i)\,\log_2 \hat{r}_\ell(i)\,,$$

where $\hat{r}_\ell(i) = r_\ell(i)/\sum_{i'} r_\ell(i')$ is the normalized attention score. The normalized entropy $H_\ell/\log_2 K$ ranges from 0 (peaked curve) to 1 (flat curve). We report this at the GAR layer for each K. Overall is the mean of the three K columns.

We use Overall (mean across K=5, 10, 15) as the reference. Gemma has the lowest entropy (65.7%), consistent with its largest GAR gain vs fixed local layout in Table 4 (+3.67 pp averaged across the three K settings for Gemma). Llama and Qwen3 show flatter profiles (96.4% and 86.9%), leaving less headroom

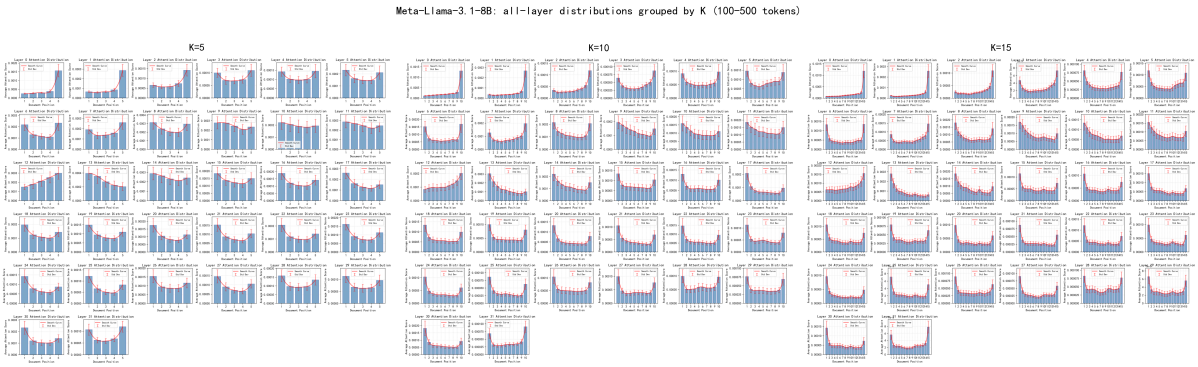

Figure 3: Meta-Llama-3.1-8B grouped all-layer attention distributions (K=5/10/15)

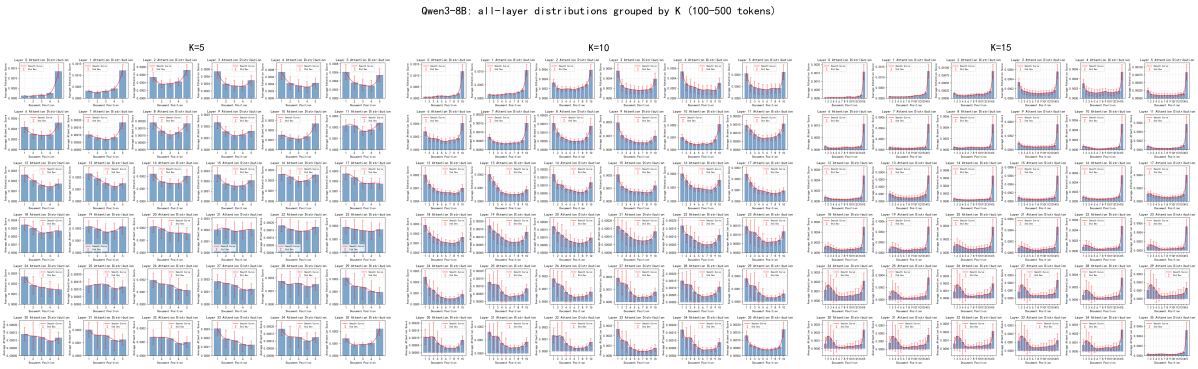

Figure 4: Qwen3-8B grouped all-layer attention distributions (K=5/10/15)

for rank–rank vs identity placement; their margins vs similarity descending are smaller. Low entropy is one factor in headroom, not a guarantee.

Layer choice. For Gemma-3-4B-IT, the final layer uses sliding-window attention only; we take $r_\ell(i)$ from a global-attention layer (layer 30). All other models use the last full-attention layer. See Appendix B for full layer-wise analysis.

B.2 B.1 Per-layer positional curves

This appendix collects all transformer layers ℓ for each of the four checkpoints from Figure 1, at three retrieval budgets ($K{=}5, 10, 15$). Figure 1 in the main text shows only the single layer per model that GAR uses (last full-attention, or global layer 30 for Gemma-3). Here, every ℓ is plotted so readers can compare early vs. deep $r_\ell(i)$ shapes and verify the last-layer choice.

B.3 B.2 Meta-Llama-3.1-8B-Instruct

Figure B1. Meta-Llama-3.1-8B all-layer positional curves, grouped by $K{=}5, 10, 15$.

B.4 B.3 Qwen3-8B

Figure B2. Qwen3-8B all-layer positional curves, grouped by $K{=}5, 10, 15$.

B.5 B.4 Gemma-3-4B-IT

Figure B3. Gemma-3-4B-IT all-layer positional curves, grouped by $K{=}5, 10, 15$.

B.6 B.5 Phi-4-mini-instruct

Figure B4. Phi-4-mini-instruct all-layer positional curves, grouped by $K{=}5, 10, 15$.

Observation. Across all models and K values, the last-layer (or global-attention layer for Gemma) consistently exhibits the strongest positional preference signal, supporting GAR's layer selection strategy.

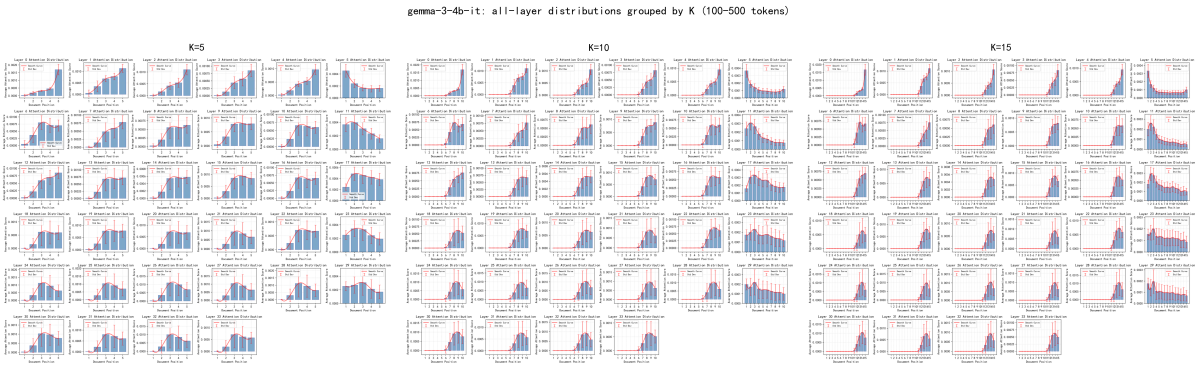

Figure 5: Gemma-3-4B-IT grouped all-layer attention distributions (K=5/10/15)

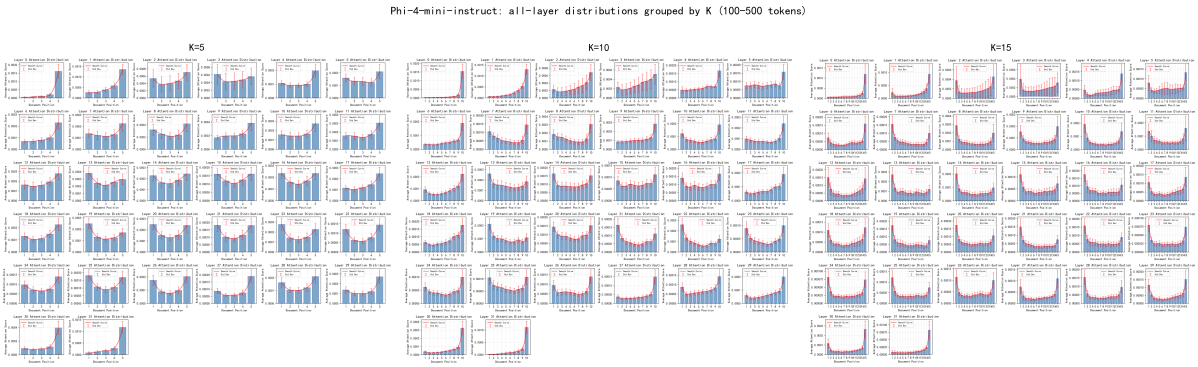

Figure 6: Phi-4-mini-instruct grouped all-layer attention distributions (K=5/10/15)

Early layers show flatter curves with less slot-to-slot variation, while deeper layers develop sharper positional profiles.

Appendix C Statistical Significance and Per-Sample Analysis

C.1 C.1 Per-condition sample sizes

All models share the same evaluation split (MuSiQue-derived, loaded then quality-filtered). Sample counts below reflect the number of items that received a valid LLM-judge response across all five conditions. Skipped items (failed extraction or generation) are excluded uniformly across conditions.

Table C1. Evaluated sample counts per model and hop stratum (K=10, GAR configuration, § 5.5).

Model	2-hop	3-hop	4-hop	Total	Skipped (loaded - evaluated)
Meta-Llama-3.1-8B-Instruct	981	989	965	2,935	19 / 11 / 35
Qwen3-8B	981	989	965	2,935	19 / 11 / 35
Gemma-3-4B-IT	981	989	965	2,935	19 / 11 / 35
Phi-4-mini-instruct	981	989	965	2,935	19 / 11 / 35

Note: the 2-hop and 3-hop effective sample sizes for paired contrasts involving local layout are smaller than the totals above because those hop strata share some samples across the H3 similarity-ranked runs (§ 5.5), reducing the number of samples with non-null entries in both conditions (e.g., Llama 2-hop paired $N \approx 400$ for GAR vs local).

C.2 C.2 Full Per-Hop Breakdowns for Tables 1–4

Table C2.1. Main comparison: GAR vs baselines (per-hop accuracy).

Model	K	Hop	GAR	AAS	Local	Desc	Asc	MidLow
Llama	5	2	0.9490	0.9195	0.9185	0.9378	0.9348	0.9348
Llama	5	3	0.9424	0.8918	0.8979	0.9434	0.9393	0.9464
Llama	10	2	0.9327	0.9195	0.9052	0.9225	0.9164	0.9225
Llama	10	3	0.8797	0.9403	0.9262	0.8726	0.8777	0.8787
Llama	10	4	0.8528	0.8487	0.8435	0.8394	0.8383	0.8456

Model	K	Hop	GAR	AAS	Local	Desc	Asc	MidLow
Llama	15	2	0.9276	0.9042	0.9001	0.9215	0.9307	0.9083
Llama	15	3	0.9393	0.9262	0.9130	0.9211	0.9343	0.9272
Llama	15	4	0.8819	0.8580	0.8477	0.8446	0.8352	0.8642
Qwen3	5	2	0.9582	0.9592	0.9531	0.9541	0.9551	0.9501
Qwen3	5	3	0.9424	0.9373	0.9282	0.9474	0.9484	0.9373
Qwen3	10	2	0.9582	0.9501	0.9511	0.9511	0.9460	0.9388
Qwen3	10	3	0.9505	0.9211	0.9333	0.9393	0.9343	0.9333
Qwen3	10	4	0.8736	0.8642	0.8798	0.8694	0.8736	0.8539
Qwen3	15	2	0.9582	0.9460	0.9511	0.9490	0.9409	0.9388
Qwen3	15	3	0.9353	0.9242	0.9282	0.9363	0.9252	0.9211
Qwen3	15	4	0.8964	0.8466	0.8943	0.8891	0.8653	0.8632
Gemma	5	2	0.9154	0.9154	0.9185	0.9083	0.9327	0.9256
Gemma	5	3	0.9242	0.9019	0.8888	0.9242	0.9100	0.9181
Gemma	10	2	0.9327	0.9062	0.9123	0.9276	0.9144	0.9001
Gemma	10	3	0.9252	0.9110	0.8787	0.9242	0.9161	0.8938
Gemma	10	4	0.8539	0.8363	0.7565	0.8601	0.8135	0.8052
Gemma	15	2	0.9256	0.9032	0.9246	0.9215	0.9144	0.9001
Gemma	15	3	0.9201	0.8918	0.8635	0.9191	0.9120	0.8888
Gemma	15	4	0.8207	0.8301	0.7606	0.8352	0.8332	0.8000
Phi	5	2	0.9348	0.9123	0.9286	0.9358	0.9388	0.9174
Phi	5	3	0.9262	0.9232	0.9292	0.9242	0.9262	0.9201
Phi	10	2	0.9337	0.8950	0.9195	0.9256	0.9235	0.9134
Phi	10	3	0.9272	0.9130	0.9201	0.9252	0.9292	0.9009
Phi	10	4	0.8539	0.8435	0.8435	0.8435	0.8570	0.8332
Phi	15	2	0.9164	0.8950	0.9103	0.9083	0.9134	0.8981
Phi	15	3	0.9312	0.9080	0.9141	0.9171	0.9171	0.9242
Phi	15	4	0.8653	0.8301	0.8363	0.8373	0.8477	0.8290

Table C2.2. H1: Middle-at-lowest vs Fixed local layout (per-hop accuracy).

Model	K	Hop	MidLow	Local	Δ
Llama	5	2	0.9144	0.9185	-0.41
Llama	5	3	0.8817	0.8979	-1.62
Llama	10	2	0.9215	0.9052	+1.63
Llama	10	3	0.9383	0.9262	+1.21
Llama	10	4	0.8456	0.8435	+0.21
Llama	15	2	0.9093	0.9001	+0.92
Llama	15	3	0.9211	0.9130	+0.81
Llama	15	4	0.8601	0.8477	+1.24
Qwen3	5	2	0.9490	0.9531	-0.41
Qwen3	5	3	0.9383	0.9282	+1.01
Qwen3	10	2	0.9409	0.9511	-1.02
Qwen3	10	3	0.9383	0.9333	+0.51
Qwen3	10	4	0.8508	0.8798	-2.90
Qwen3	15	2	0.9378	0.9511	-1.33
Qwen3	15	3	0.9201	0.9282	-0.81
Qwen3	15	4	0.8632	0.8943	-3.11
Gemma	5	2	0.9246	0.9185	+0.61
Gemma	5	3	0.9151	0.8888	+2.63
Gemma	10	2	0.9001	0.9123	-1.22
Gemma	10	3	0.8938	0.8787	+1.52
Gemma	10	4	0.8052	0.7565	+4.87
Gemma	15	2	0.9001	0.9246	-2.45
Gemma	15	3	0.8888	0.8635	+2.53
Gemma	15	4	0.8000	0.7606	+3.94
Phi	5	2	0.9174	0.9286	-1.12
Phi	5	3	0.9201	0.9292	-0.91
Phi	10	2	0.9134	0.9195	-0.61
Phi	10	3	0.9009	0.9201	-1.92
Phi	10	4	0.8332	0.8435	-1.04
Phi	15	2	0.8981	0.9103	-1.22
Phi	15	3	0.9242	0.9141	+1.01
Phi	15	4	0.8290	0.8363	-0.73

Table C2.3. H3: GAR vs baselines (per-hop Δ).

Model	K	Hop	Δ(GAR-AAS)	Δ(GAR-Local)	Δ(GAR-Desc)	Δ(GAR-Asc)
Llama	5	2	+2.96	+2.24	+1.12	+1.43
Llama	5	3	+5.06	+0.71	-0.10	+0.30
Llama	10	2	+1.33	+0.92	+1.02	+1.63
Llama	10	3	-6.07	-0.20	+0.71	+0.20
Llama	10	4	+0.41	+0.93	+1.35	+1.45
Llama	15	2	+2.34	+1.02	+0.61	-0.31
Llama	15	3	+1.31	+1.82	+1.82	+0.51
Llama	15	4	+2.38	+3.73	+3.73	+4.66
Qwen3	5	2	-0.10	+0.61	+0.41	+0.31
Qwen3	5	3	+0.51	+0.51	-0.51	-0.61
Qwen3	10	2	+0.82	+0.41	+0.71	+1.22
Qwen3	10	3	+2.93	+1.52	+1.11	+1.62
Qwen3	10	4	+0.93	+0.73	+0.41	+0.00
Qwen3	15	2	+1.22	+0.10	+0.92	+1.73
Qwen3	15	3	+1.11	-0.10	-0.10	+1.01
Qwen3	15	4	+4.97	+0.73	+0.73	+3.11
Gemma	5	2	+0.00	-1.12	+0.71	-1.73
Gemma	5	3	+2.22	+3.24	+0.00	+1.42
Gemma	10	2	+2.65	+0.51	+0.51	+1.83
Gemma	10	3	+1.42	+3.74	+0.10	+0.91
Gemma	10	4	+1.76	+9.22	-0.62	+4.04
Gemma	15	2	+2.24	-0.51	+0.41	+1.12
Gemma	15	3	+2.83	+3.74	+0.10	+0.81
Gemma	15	4	-0.93	+5.60	-1.45	-1.24
Phi	5	2	+2.24	-1.02	-0.10	-0.41
Phi	5	3	+0.30	-0.61	+0.20	+0.00
Phi	10	2	+3.87	+0.41	+0.82	+1.02
Phi	10	3	+1.42	+0.20	+0.20	-0.20
Phi	10	4	+1.04	+1.66	+1.04	-0.31
Phi	15	2	+2.14	+0.31	+0.82	+0.31
Phi	15	3	+2.33	+1.42	+1.42	+1.42
Phi	15	4	+3.52	+1.45	+2.80	+1.76

C.3 C.3 Bootstrap 95% confidence intervals for Tables 2–3 (K=10)

We compute paired bootstrap CIs (10,000 resamples) for the per-stratum Δ values in Tables 2 (H1) and 3 (H2). An asterisk (*) indicates that the 95% CI does not include zero, i.e., the difference is significant at α=0.05.

Table C3.1. H1: bootstrap 95% CI for Δ=GAR - baseline (K=10, GAR configuration), by generator and hop stratum. Values in percentage points (×100).

Model	Contrast	Δ (2-hop) 95% CI	Δ (3-hop) 95% CI	Δ (4-hop) 95% CI	Overall 95% CI
Llama	GAR - local	[+0.75; -2.00, +3.49]	[+0.00; -3.53, +3.76]	[+0.93; -1.55, +3.42]	[+0.67; -1.06, +2.40]
Llama	GAR - desc.	[+1.25; -1.50, +3.99]	[-0.47; -3.76, +2.82]	[+1.35; -1.04, +3.73]	[+0.89; -0.78, +2.51]
Qwen3	GAR - local	[+0.41; -0.92, +1.63]	[+1.52; -0.10, +3.03]	[-3.79; -6.75, -0.99]*	[-0.16; -1.16, +0.89]
Qwen3	GAR - desc.	[+0.71; -0.31, +1.73]	[+1.11; -0.20, +2.43]	[+0.00; -2.47, +2.47]	[+0.70; -0.16, +1.55]
Gemma	GAR - local	[+3.77; +0.00, +7.55]	[+3.74; +1.62, +5.97]*	[+9.22; +6.32, +12.12]*	[+6.19; +4.48, +7.89]*
Gemma	GAR - desc.	[+0.47; -0.94, +2.36]	[+0.10; -0.91, +1.11]	[-0.62; -2.07, +0.93]	[-0.18; -1.02, +0.65]
Phi	GAR - local	[+0.41; -1.22, +2.04]	[+0.90; -1.01, +2.69]	[+1.66; -0.83, +4.04]	[+0.99; -0.18, +2.15]
Phi	GAR - desc.	[+0.82; -0.82, +2.34]	[+0.34; -1.35, +2.02]	[+1.04; -1.24, +3.32]	[+0.74; -0.32, +1.83]

Table C3.2. H2: bootstrap 95% CI for Δ=GAR - middle-at-lowest (K=10, GAR configuration).

Model	Δ (2-hop) 95% CI	Δ (3-hop) 95% CI	Δ (4-hop) 95% CI
Llama	[-0.50; -3.00, +2.25]	[+1.18; -2.36, +4.72]	[+0.73; -1.76, +3.21]
Qwen3	[+1.94; +0.51, +3.47]*	[+1.72; +0.20, +3.24]*	[+1.32; -1.48, +4.12]
Gemma	[+6.60; +2.36, +11.32]*	[+2.93; +0.71, +5.06]*	[+5.18; +2.28, +8.08]*

Model	Δ (2-hop) 95% CI	Δ (3-hop) 95% CI	Δ (4-hop) 95% CI
Phi	[+2.14; +0.51, +3.87]*	[+2.69; +0.79, +4.60]*	[+2.49; +0.00, +4.97]

C.4 C.4 Bootstrap 95% CIs for Table 7 (robustness across K)

Table C4. Overall Δ (pp, ×100) with 95% bootstrap CIs for GAR vs baselines across K=5, 10, 15. Note: K=5 pooled over 2/3-hop only (Llama: 3-hop only); K=10, 15 pooled over 2/3/4-hop.

Model	K	GAR - local CI	GAR - desc. CI	GAR - mid-low CI
Llama	5†	—	—	—
Llama	10	[+0.67; -1.06, +2.35]	[+0.89; -0.78, +2.51]	[+0.56; -1.12, +2.29]
Llama	15	[+2.24; +0.66, +3.82]*	[+1.38; +0.00, +2.76]	[+1.25; -0.26, +2.76]
Qwen3	5	[+0.56; -0.43, +1.55]	[+0.03; -0.91, +0.97]	[+0.66; +0.20, +1.12]*
Qwen3	10	[-0.16; -1.13, +0.89]	[+0.70; -0.16, +1.55]	[+1.71; +0.66, +2.72]*
Qwen3	15	[+0.45; -0.54, +1.44]	[+0.52; -0.42, +1.46]	[+2.33; +1.82, +2.84]*
Gemma	5	[+1.06; +0.12, +2.00]*	[+0.04; -0.84, +0.92]	[-0.02; -0.90, +0.86]
Gemma	10	[+6.19; +4.48, +7.89]*	[-0.18; -1.02, +0.65]	[+4.29; +2.59, +6.00]*
Gemma	15	[+2.94; +1.60, +4.26]*	[-0.31; -1.33, +0.72]	[+2.38; +0.95, +3.82]*
Phi	5	[-0.81; -1.86, +0.24]	[+0.25; -0.69, +1.19]	[+1.12; +0.54, +1.70]*
Phi	10	[+0.99; -0.14, +2.15]	[+0.74; -0.35, +1.83]	[+2.43; +1.23, +3.63]*
Phi	15	[+1.06; +0.10, +2.02]*	[+1.68; +0.72, +2.64]*	[+2.02; +1.08, +2.96]*

† Llama K=5 CI not computed (2-hop data corrupted, single-hop CI unreliable).

C.5 C.5 Per-sample GAR gain distribution

To characterize the granularity of GAR's effect, we compute per-sample accuracy differences (GAR - fixed local) at K=10 under the GAR configuration in § 5.5. Because each sample produces a binary correct/incorrect judgment, most individual deltas are 0, +1, or -1 (in 0/1 scale), and the mean over many such deltas yields the aggregate Δ in Tables 5–7.

Table C5. Per-sample GAR gain distribution (K=10, GAR configuration, pooled over 2/3/4-hop).

Model	N	Mean (pp)	Std (pp)	GAR > Fixed	GAR < Fixed	GAR = Fixed
Llama	1,791	+0.67	37.1	7.2%	6.5%	86.3%
Qwen3	2,577	-0.16	26.4	3.4%	3.6%	93.0%
Gemma	2,166	+6.19	40.2	11.4%	5.2%	83.5%
Phi	2,837	+0.99	31.3	5.4%	4.4%	90.2%

The high fraction of ties (GAR = Fixed) reflects the binary judgment at high baseline accuracy: when both conditions answer correctly (or both incorrectly) on the same item, the per-sample delta is zero. Gemma shows the largest "win fraction" (11.4%) and the largest mean gain, consistent with its peaked attention profile (Figure 1). Cross-model per-sample gain correlations are near zero (Pearson r < 0.08 for all pairs), indicating that GAR's benefit is question-specific and not driven by a shared subset of "easy" or "hard" items.

Appendix D Proofs and Additional Details

D.1 D.1 Collapsed column formula

When each block is represented by a single key column i (as in our tabular layout), the per-slot attention in § 2 reduces to mean sink attention into column i:

$$r_\ell(i) \;=\; \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\frac{1}{L}\sum_{j=1}^{L} A_c^{(\ell)}(j,i)\,.$$

(Optional variants: query-only aggregation restricts the sum to question positions; normalization $\hat{r}_\ell(i) = r_\ell(i)/\sum_{i'} r_\ell(i')$ enables cross-layer comparison. GAR uses the same slot ranking in all cases.)

D.2 D.2 Chain vs. prompt order

The logical multi-hop sequence is encoded in block content and relations. Left-to-right prompt order after GAR need not follow $b_1, \ldots, b_K$: GAR permutes slots, not the underlying hop facts. What is preserved under relation-chain ordering is the identity of each b_j as the j-th hop block—not that b_1 appears to the left of b_K in the string (cf. H3 vs. similarity-ranked blocks in § 5.5).

D.3 D.3 Segment-wise assignment

With multiple segments, we use segment-wise assignment (§ 4) so that within-segment chain order is preserved while slot multisets follow σ.

D.4 D.4 Remarks on hypotheses

(i) H1 stresses salience for intermediates; H2 stresses σ; H3 stresses relation-chain vs. similarity-based block order as inputs to rank–rank matching.

(ii) Architectures with sliding-window attention complicate "last layer" statistics; we use the last full-attention layer or a fixed global layer when the top layer is local-only (e.g., Gemma 3).

Appendix E Concrete Example of H1 (Chain Breakdown)

To illustrate H1, we present a 3-hop sample where middle-at-lowest fails while GAR succeeds. This demonstrates that placing intermediate chain nodes in low-attention regions breaks reasoning even when head and tail nodes remain attended.

E.1 E.1 Setup

Question: What is the main international airport in the place of death of John III, former ruler of the country where Immanu El was formed?

Reference Answer: Stockholm Arlanda Airport

Answer chain: Immanu El → Sweden → Stockholm → Stockholm Arlanda Airport

E.2 E.2 Middle-at-lowest (Incorrect)

Slot assignment: Nodes 0 (Immanu El) and 3 (Airport) in high-attention slots; Nodes 1–2 (Sweden, Stockholm) in lowest-attention slots.

Model prediction (Gemma-3-4B-IT, K=10):

> [...] 1. Immanu El was formed in Sweden: [...] 2. Stockholm is in Sweden: [...] 3. Stockholm Arlanda Airport is in Stockholm: [...] 4. John III ruled Sweden: We need to find information about John III's place of death. Unfortunately, none of the provided nodes contain this information.
>
> Therefore, we cannot determine the main international airport in the place of death of John III.

Failure: The model identifies the chain from distractor nodes but fails to retrieve "John III died in Stockholm" because intermediate nodes are in low-attention slots. The reasoning breaks at John III → Stockholm.

E.3 E.3 GAR (Correct)

Slot assignment: All four chain nodes in high-attention slots (Algorithm 1).

Model prediction (Gemma-3-4B-IT, K=10):

> [...] 1. Immanu El was formed in Sweden: [...] 2. Stockholm is located in Sweden: [...] 3. Stockholm is the place of death: Node 3 states this. 4. Stockholm Arlanda Airport is near Stockholm: [...]
>
> Therefore, the main international airport in the place of death of John III is Stockholm Arlanda Airport.

Success: With intermediate nodes (Sweden, Stockholm) in high-attention slots, the model retrieves "Stockholm is the place of death" and completes the chain.

E.4 E.4 Interpretation

This example supports H1: intermediate chain nodes require attended slots for reliable chain following. Even when head and tail nodes are attended and chain order is preserved, placing intermediate nodes in low-attention slots breaks reasoning. GAR prevents this by assigning all chain nodes to attended slots.

Appendix F HotpotQA Robustness Results

Using the same entity-grounded ordering protocol on HotpotQA [32], we evaluate GAR against Local/Descending/Ascending baselines on the `Overall` split reported in the project summary workbook (`HotpotQA_EntitySim` sheet). This provides an out-of-domain robustness check complementary to the MuSiQue-derived main evaluation.

Table F1. HotpotQA overall accuracy (%) and GAR deltas (pp) for available K settings in the summary sheet.

Model	K	GAR	Local	Desc	Asc	Δ(GAR-Local)	Δ(GAR-Desc)	Δ(GAR-Asc)
Gemma	5	79.37	77.76	80.24	80.36	+1.61	-0.87	-0.99
Gemma	10	81.91	78.81	81.72	80.05	+3.10	+0.19	+1.86
Llama	5	80.24	79.55	79.49	80.24	+0.69	+0.75	+0.00
Llama	10	82.34	80.85	81.91	81.97	+1.49	+0.43	+0.37
Phi	5	82.78	81.91	81.60	83.52	+0.87	+1.18	-0.74
Phi	10	82.84	82.46	81.54	82.71	+0.38	+1.30	+0.13
Qwen3	5	84.76	84.57	84.82	84.95	+0.19	-0.06	-0.19
Qwen3	10	85.87	86.55	85.63	86.25	-0.68	+0.24	-0.38

Across these 8 model–K cells, GAR is above Local in 7/8 cells (mean +0.96 pp), above Desc in 5/8 cells (mean +0.40 pp), and above Asc in 4/8 cells (mean +0.01 pp), indicating that the attention-aligned placement signal transfers beyond the main MuSiQue-derived set but remains model- and baseline-dependent.

F.1 F.1 HotpotQA per-hop breakdown

For completeness, we also report the per-hop rows (2-hop, 3-hop, 4-hop) from the same `HotpotQA_EntitySim` sheet for $K{=}5, 10$.

Table F2. HotpotQA per-hop accuracy (%) and GAR deltas (pp).

Model	K	Hop	GAR	Local	Desc	Asc	Δ(GAR-Local)	Δ(GAR-Desc)	Δ(GAR-Asc)
Gemma	5	2-hop	80.04	79.61	81.02	82.32	+0.43	-0.98	-2.28
Gemma	5	3-hop	78.90	76.13	79.49	77.71	+2.77	-0.59	+1.19
Gemma	5	4-hop	77.30	72.97	78.38	77.84	+4.33	-1.08	-0.54
Gemma	10	2-hop	82.75	81.67	82.75	81.34	+1.08	+0.00	+1.41
Gemma	10	3-hop	81.66	75.15	81.46	79.88	+6.51	+0.20	+1.78
Gemma	10	4-hop	78.38	74.59	77.30	74.05	+3.79	+1.08	+4.33
Llama	5	2-hop	80.26	80.80	79.50	80.59	-0.54	+0.76	-0.33
Llama	5	3-hop	81.07	78.50	81.07	81.66	+2.57	+0.00	-0.59
Llama	5	4-hop	77.84	76.22	75.14	74.59	+1.62	+2.70	+3.25
Llama	10	2-hop	83.95	82.21	83.41	83.41	+1.74	+0.54	+0.54
Llama	10	3-hop	81.66	79.68	80.28	80.28	+1.98	+1.38	+1.38
Llama	10	4-hop	76.22	77.30	78.92	79.46	-1.08	-2.70	-3.24
Phi	5	2-hop	83.95	83.84	82.54	84.06	+0.11	+1.41	-0.11
Phi	5	3-hop	80.87	79.09	80.08	82.05	+1.78	+0.79	-1.18
Phi	5	4-hop	82.16	80.00	81.08	84.86	+2.16	+1.08	-2.70
Phi	10	2-hop	83.84	83.51	82.86	83.84	+0.33	+0.98	+0.00
Phi	10	3-hop	81.66	80.87	80.08	80.87	+0.79	+1.58	+0.79
Phi	10	4-hop	81.08	81.62	78.92	82.16	-0.54	+2.16	-1.08
Qwen3	5	2-hop	85.68	85.90	85.79	86.23	-0.22	-0.11	-0.55
Qwen3	5	3-hop	83.43	82.64	83.04	83.04	+0.79	+0.39	+0.39
Qwen3	5	4-hop	83.78	83.24	84.86	83.78	+0.54	-1.08	+0.00
Qwen3	10	2-hop	87.74	88.50	87.53	87.64	-0.76	+0.21	+0.10
Qwen3	10	3-hop	83.83	84.81	84.02	84.62	-0.98	-0.19	-0.79
Qwen3	10	4-hop	82.16	81.62	80.54	83.78	+0.54	+1.62	-1.62

At per-hop granularity, GAR remains model- and hop-dependent on HotpotQA: gains are strongest for Gemma, mixed for Llama/Phi by hop depth, and near-neutral on Qwen3 overall. This pattern is consistent with the model-specific positional-geometry discussion in § 6.