What Structural Inductive Bias Helps Transformers Reason Over Knowledge Graphs?

A Study with Tabula RASA

Jonas Petersen [1 2] **Camilla Mazzoleni** [1] **Gian-Alessandro Lombardi** [1] **Federico Martelli** [1] **Riccardo Maggioni** [1 3]

Abstract

What structural inductive bias helps transformers reason over knowledge graphs? Through controlled ablations of a minimal transformer modification with four independently removable components (sparse adjacency masking, edge-type biases, query scaling, value gating), we isolate which structural signals drive multi-hop reasoning. Our finding is sharp: sparse adjacency masking alone accounts for the dominant share of improvement over unmasked transformers (+72.5pp on 3-hop MetaQA, +45.5pp on WebQSP, +53.9pp on CWQ), while learned relation parameters add only modest refinement and can actively hurt without structural guidance. A zero-shot experiment provides architecturally independent corroboration: masking-based attention degrades 4.0× less than relation-specific weights when edge types are held out. The useful inductive bias for multi-hop KGQA is predominantly topological, not relational.

1. Introduction

Transformers (Vaswani et al., 2017) have reshaped language, vision, and code, yet they remain fragile on tasks that demand systematic relational reasoning, following chains of typed relationships through structured data such as knowledge graphs (Ren et al., 2020; Zhang et al., 2022; Press et al., 2023; Dziri et al., 2023). A sizeable literature has responded by injecting graph structure into the attention mechanism: centrality and spatial encodings (Ying et al., 2021), spectral positional features (Kreuzer et al., 2021), hybrid local-global attention (Rampašek et al., 2022), and structure-aware preprocessing (Chen et al., 2022). Each method, however, couples several design choices, making it difficult to tell which ingredient is carrying the performance improvement.

The question this paper investigates is not "how do we design a better graph transformer?" but rather: *among the structural signals a transformer can be given about a knowledge graph, which ones actually matter, and by how much?* In the spirit of recent work that revisits architectural folklore empirically (Brody et al., 2022; Tönshoff et al., 2023; Ying et al., 2021), we treat graph transformer design as a controlled experiment.

To do so we construct a minimal vehicle, RASA (**R**elation-**A**ware **S**parse **A**ttention), in which four candidate structural signals are exposed as *independently removable* components: (i) a binary adjacency mask that restricts each layer's attention to graph neighbors, (ii) a learnable edge-type bias b_r, (iii) a relation-specific query scale s_r, and (iv) a relation-specific value gate g_r. Each component can be switched on or off without retraining the encoder, enabling a clean ablation over the space of structural biases. RASA itself is not the contribution; it is the experimental apparatus.

Our headline finding is sharp: across three KGQA benchmarks (MetaQA, WebQSP, CWQ) and two generalization experiments, the dominant factor is almost always the binary adjacency mask. Masking alone recovers the overwhelming majority of the full model's improvement; on CWQ, learned biases without masking perform *below* the vanilla transformer (2.1% vs. 2.7%). This suggests the useful inductive bias is predominantly *topological*, not *relational*. A depth analysis (Section 3) explains why: each attention layer extends a node's receptive field by at most one hop, so k-hop reasoning requires k layers. Masking directly enforces this propagation, whereas learned relation parameters must rediscover it from data.

Contributions. We make two insight-centric contributions:

1. **Adjacency masking dominates learned relation parameters.** Through controlled ablations on three KGQA benchmarks, we quantify the contribution of adjacency masking (Dwivedi & Bresson, 2021) against three learned relational signals (edge-type biases, query scaling, value gating) and find that the former domi-

[1]Forgis [2]University of Cambridge [3]ETH Zurich. Correspondence to: Jonas Petersen <jep79@cantab.ac.uk>.

Workshop on Graph Foundation Models at the 43rd International Conference on Machine Learning (ICML 2026), Seoul, South Korea. Copyright 2026 by the author(s).

nates: masking alone accounts for the overwhelming majority of the improvement over an unmasked transformer (+72.5pp on 3-hop MetaQA, +45.5pp on WebQSP, +53.9pp on CWQ). Learned relation parameters add modest refinement and, without masking, can actively harm performance.

2. **Structural masking generalizes where relation-specific weights do not.** In a zero-shot experiment in which edge types are held out at training time, structural-masking attention degrades $4.0\times$ less than R-GCN's relation-specific weight matrices (-7.2pp vs. -29.2pp), providing a second, architecturally independent line of evidence that the primary useful bias for multi-hop KGQA is topological.

The configuration we evaluate is competitive with graph-native baselines (92.6% on 3-hop MetaQA, 72.5% on WebQSP, 59.9% on CWQ), confirming that the ablation findings are not an artifact of a weak baseline. We explicitly do not claim state-of-the-art: LLM-augmented methods such as SubgraphRAG (Li et al., 2024) reach 90.1% on WebQSP and operate in a different regime. Our claim concerns *what matters* among pure structural biases, not a leaderboard win.

2. Related Work

Depth and complexity of transformers. Circuit-complexity results show that fixed-depth transformers with bounded precision compute functions in TC^0 (Merrill & Sabharwal, 2022; 2023), and that they cannot solve graph connectivity at constant depth (Sanford et al., 2023). Empirically, transformers fail on compositional multi-step tasks as reasoning depth grows (Dziri et al., 2023; Press et al., 2023). This motivates the depth argument in Section 3: k-hop reasoning requires $\Omega(k)$ layers, and architectural choices that directly enforce one-hop-per-layer propagation are therefore prior candidates for dominant structural signals.

Graph transformers and structured attention. Most relevant to our Component 1, Dwivedi & Bresson (2021) introduced sparse graph transformers with adjacency-based attention masking. Our work does not re-introduce this mechanism; we use it as one arm of a controlled ablation to quantify its contribution relative to learned relation parameters in the multi-hop KGQA setting. Graphormer (Ying et al., 2021) takes a different approach, using dense attention with spatial and centrality encodings; SAN (Kreuzer et al., 2021) uses Laplacian-based positional encodings; GraphGPS (Rampašek et al., 2022) combines message passing with global attention; SAT (Chen et al., 2022) augments attention with GNN-extracted features; Exphormer (Shirzad et al., 2023) sparsifies attention using expander graphs and virtual nodes (our approach is simpler, adjacency-only, with no expander or virtual node augmentation, and our ablation quantifies the mechanism's individual contribution against learned relation parameters); and Ma et al. (2023) learn to combine local and global attention adaptively. These works target molecular and node-level tasks and typically bundle several structural signals together, making individual-component isolation difficult. Our ablation is specifically designed to isolate the topological-vs-relational split for multi-hop KGQA. Sequence-level sparse attention (Longformer, BigBird, ETC) is related but relies on fixed patterns; RASA's sparsity is derived from the input graph structure.

Knowledge graph QA. KGQA methods range from embedding-based scorers (EmbedKGQA (Saxena et al., 2020), NSM (He et al., 2021)), to GNN+LM hybrids (QA-GNN (Yasunaga et al., 2021), GreaseLM (Zhang et al., 2022), ReaRev (Mavromatis et al., 2022)), to semantic parsing (Berant et al., 2013) and retrieval-augmented LLM pipelines (RoG (Luo et al., 2024), GNN-RAG (Mavromatis & Karypis, 2024), SubgraphRAG (Li et al., 2024)). LLM-augmented methods are state-of-the-art on WebQSP/CWQ but leverage external text knowledge; we operate in the purely graph-structural regime to keep the architectural question sharp. Among graph-native baselines we compare against R-GCN (Schlichtkrull et al., 2018), GAT (Veličković et al., 2018), and Graphormer, all with a matched frozen DistilBERT encoder.

3. Why Depth, and Hence Masking, Matters

Our ablation findings (Section 5) are best understood through a simple depth argument: k-hop reasoning requires at least k transformer layers, and among the four candidate structural signals, only adjacency masking directly enforces the one-hop-per-layer propagation this argument demands. The rest of this section formalizes this intuition; the full complexity-theoretic background is deferred to Appendix A.

Definition 3.1 (k-hop Relational Reasoning)**.** Given a knowledge graph $G = (V, E, R)$ with vertices V, typed edges $E \subseteq V \times R \times V$, and relation types R, a k-hop query asks: starting from entity e_0, following relations $(r_1, \ldots, r_k)$, which entities can be reached?

Theorem 3.2 (Depth Required for k-hop)**.** *Any transformer computing k-hop reachability for k growing with the input size requires $\Omega(k)$ layers, even under global attention.*

Proof sketch. Initialize each node with only its own identity embedding. Attention is a weighted average of value vectors, so a single layer aggregates each node's direct neighbors but cannot *compose* paths: B's representation at layer 1 contains its own features and C's features, but not a composed "$A \to B \to C$" representation. Hence each layer extends a node's effective receptive field by exactly one hop, and k-hop reachability requires $\Omega(k)$ layers. A reduction to graph connectivity on layered graphs, together with

the observation that connectivity lies outside TC^0 (Barrington et al., 1990; Merrill & Sabharwal, 2023; Sanford et al., 2023), gives the formal bound; see Appendix A. □

For the fixed $k \in \{1, 2, 3\}$ in our experiments, k-hop reachability is in TC^0 (constant-depth circuits can compute A^k), so the lower bound does not apply as a complexity-theoretic obstruction. What *does* apply uniformly is the information-propagation argument in the sketch: each attention layer extends a node's effective receptive field by at most one hop. Figure 6 illustrates this on a linear chain.

RASA as a structural instantiation. RASA attention is the standard scaled dot-product operation modified with four independently removable components:

$$S_{ij} = \frac{(XW_Q)_i \cdot (XW_K)_j}{\sqrt{d_k}} + b_{E_{ij}} \tag{1}$$

$$S_{ij} \leftarrow S_{ij} \cdot s_{E_{ij}} \tag{2}$$

$$\tilde{S}_{ij} = \begin{cases} S_{ij} & \text{if } A_{ij} = 1 \text{ or } i = j \\ -\infty & \text{otherwise} \end{cases} \tag{3}$$

$$\mathrm{Attn}(X) = \big(\mathrm{softmax}(\tilde{S}) \odot \sigma(\mathbf{g}_E)\big) XW_V \tag{4}$$

where $b_r, s_r, g_r \in \mathbb{R}^H$ are learnable per-relation parameters. The key property is that the mask in (3) makes each layer's aggregation *exactly* one graph hop; a global-attention transformer, even given the same edge-type biases, has to learn this behavior from data.

Remark 3.3 (Search-space intuition). For a graph with n nodes and m edges, the space of binary attention patterns available to a standard transformer has size $O(2^{n^2})$; masking restricts it to $O(2^m)$. For sparse graphs ($m = O(n)$) this is an exponential reduction. Transformers learn continuous attention weights rather than binary masks, so this is an intuition rather than a formal sample-complexity bound, but it motivates why masking-induced inductive bias should simplify learning relative to a purely learned signal.

Theory predicts masking dominance. Taken together, these observations yield a concrete prediction for the ablation to come: since k-hop reasoning requires exactly k hops of one-hop-per-layer propagation, the structural signal that *directly enforces* one-hop-per-layer aggregation should dominate signals that only re-weight an already-aggregated representation. Among RASA's four components, only the adjacency mask (eq. 3) has this property: edge-type biases, query scaling, and value gating all reshape scores or outputs *after* the attention pattern is set, but they cannot by themselves restrict information flow to graph edges. We verify this prediction empirically in Section 5.3, where masking alone contributes +72.5/+45.5/+53.9pp across three benchmarks, while the three learned relation parameters together add only modest additional gains.

4. RASA: A Minimal Vehicle for Isolating Structural Bias

To isolate the effect of structural bias, we build a minimal transformer modification in which four candidate structural signals are exposed as *independently removable* components. Our goal is not to propose a new architecture but to construct an experimental apparatus that lets us turn each signal on or off in isolation and measure its individual contribution. Figure 1 summarizes the resulting attention operation.

The four components are ordered by the role we hypothesize they play (see Section 3): masking sets the *topological* attention pattern, while biases, query scaling, and gating are *relational* re-weightings layered on top of that pattern. The implementation reduces to four extra lines over standard scaled dot-product attention:

Listing 1. RASA attention (4 modifications over standard attention).

```
S = Q @ K.T / sqrt(d)
S.masked_fill_(~adj, -inf)     # 1) mask
S += bias[edge_types]          # 2) bias
S *= scale[edge_types]         # 3) scale
W = softmax(S)
W *= sigmoid(gate[edge_types]) # 4) gate
output = W @ V
```

Component 1: Sparse adjacency masking (topological). A binary mask A where $A_{ij} = 1$ if any relation connects i to j, or $i = j$, following Dwivedi & Bresson (2021). Non-adjacent positions receive $-\infty$ scores. This is the only component that restricts *which* positions can attend to which, and is therefore the only component that directly enforces one-hop-per-layer propagation (Section 3).

Component 2: Edge-type biases (relational). A learnable bias $b_r \in \mathbb{R}^H$ per relation type r and attention head, added to attention scores between connected positions.

Component 3: Relation-specific query scaling (relational). A learnable scalar $s_r \in \mathbb{R}^H$ that modulates attention scores per relation type, enabling relation-specific attention intensity.

Component 4: Relation-specific value gating (relational). A learned gate $g_r \in [0, 1]^H$ per relation type that controls how much information flows through each edge type.

We also include a Graphormer-style degree encoding held constant across ablations (see Appendix B).

These modifications require $3|R|H$ additional parameters per layer. For MetaQA ($|R| = 9$, $H = 4$), this is 108 parameters per layer for the relation components, negligible relative to the ∼66M DistilBERT encoder. Components 2-4 are never evaluated individually in our main ablation (we test them in aggregate as "relation parameters"); the central comparison is between topological masking and relational

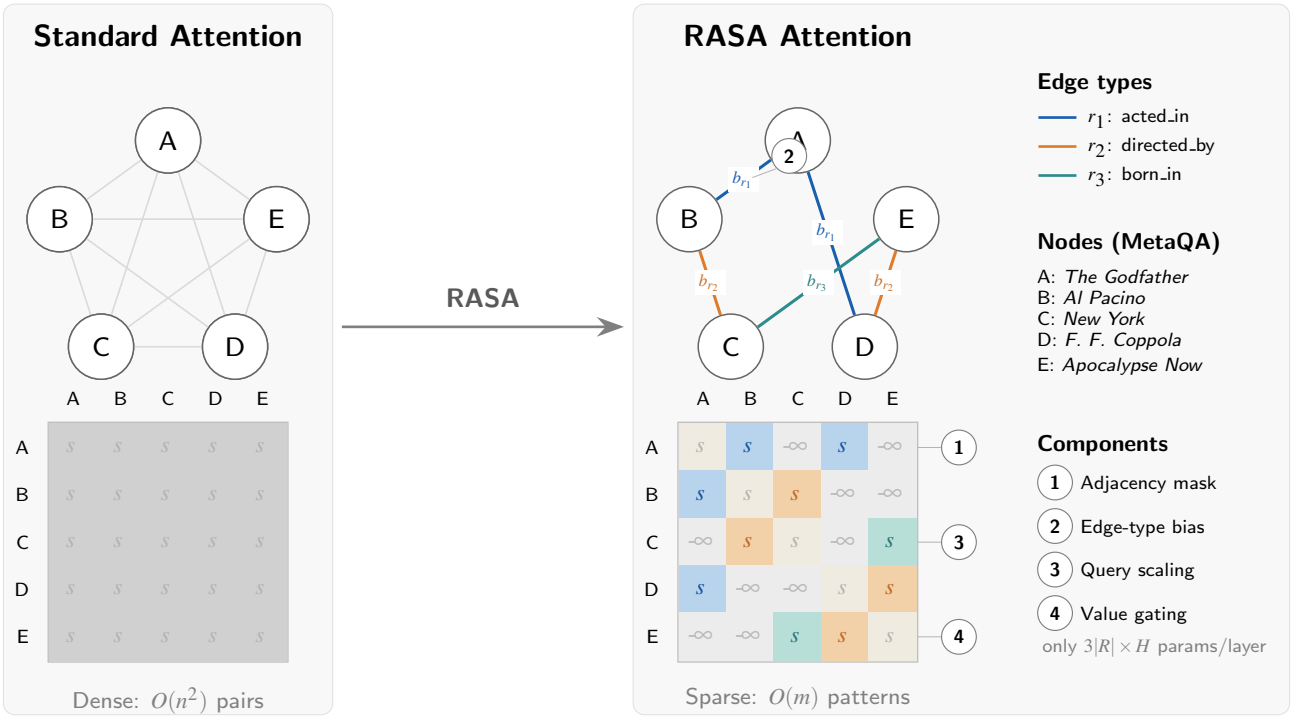

Figure 1. RASA replaces dense $O(n^2)$ attention (left) with sparse $O(m)$ relation-aware attention (right) via four components: **(1)** adjacency mask, **(2)** edge-type bias b_r, **(3)** query scaling s_r, **(4)** value gating g_r. Only $3|\mathcal{R}| \times H$ parameters per layer. Each component can be independently disabled for ablation.

re-weighting.

5. Experiments

5.1. Setup

Datasets. We evaluate on three KGQA benchmarks:

- **MetaQA** (Zhang et al., 2018): Multi-hop KGQA over a movie knowledge graph with 1/2/3-hop questions, 9 relation types, and 43K entities.
- **WebQuestionsSP** (WebQSP) (Yih et al., 2016): Real-world questions from web searches over Freebase with 21 relation types and 11.6K entities (in our Freebase subgraph).
- **ComplexWebQuestions** (CWQ) (Talmor & Berant, 2018): Compositional multi-hop reasoning over Freebase with 27K training examples.

Implementation. RASA uses a DistilBERT encoder with a 3-layer RASA transformer (128 hidden dim, 4 attention heads). Training: batch size 16, lr 2e-5, AdamW with cosine annealing, early stopping (patience 5). All experiments run on NVIDIA A10G (24GB). Full hyperparameters in Appendix B. All results report mean $\pm$ std over 3 seeds (42, 123, 456); margin analysis in Appendix K.

Baselines. We compare against: (1) **GNN baselines**: R-GCN (Schlichtkrull et al., 2018), GAT (Veličković et al., 2018); (2) **Transformer baselines**: Vanilla Transformer (full attention, no structural encodings), Graphormer (Ying et al., 2021) (full attention with centrality and spatial encodings); (3) **Published results**: EmbedKGQA (Saxena et al., 2020), NSM (He et al., 2021). All our baselines use the same DistilBERT encoder and answer scoring architecture for fair comparison. Graphormer on WebQSP/CWQ was tuned via a single-seed HP screen over $d \in \{128, 256\}$, $L \in \{3, 4, 5\}$, lr $\in \{2\times10^{-5}, 5\times10^{-5}, 10^{-4}\}$, with the best configuration reported at 500 evaluation samples on CWQ (Appendix B); this HP budget matches RASA's on the same datasets.

5.2. Main Results: Competitive, Not SOTA

Before turning to the ablation, we verify that the configuration we are ablating is a meaningful one. Table 1 and Figure 2 report MetaQA results: the full configuration reaches 92.6% on 3-hop, comparable to R-GCN (91.9%) and Graphormer (93.3%) under matched encoder and compute, while the vanilla transformer collapses to 12.9%. Notably, performance *increases* with reasoning depth for the full model (63.5% $\rightarrow$ 72.7% $\rightarrow$ 92.6%), consistent with the depth argument of Section 3. The point of this table is not a leaderboard win; it is to establish that the ablation findings

Table 1. Hits@1 (%) on MetaQA. Mean ± std over 3 seeds. Bold = best in each column. Published results (EmbedKGQA, NSM) use pre-trained KG embeddings not available to other models.

Model	1-hop	2-hop	3-hop
With pre-trained KG embeddings:			
EmbedKGQA†	97.5	98.8	94.8
NSM†	97.1	99.9	98.9
Without pre-trained KG embeddings (ours):			
Vanilla Transformer	39.8	0.8	12.9±0.2
Graphormer	94.4±0.8	74.8±0.2	**93.3**±0.2
R-GCN	**97.9**±0.2	**85.9**±1.7	91.9±0.2
GAT	49.2±1.5	74.8±1.1	86.8±1.6
RASA (ours)	63.5±6.6	72.7±1.0	92.6±0.1

†Uses pre-trained KG embeddings (ComplEx/DistMult).

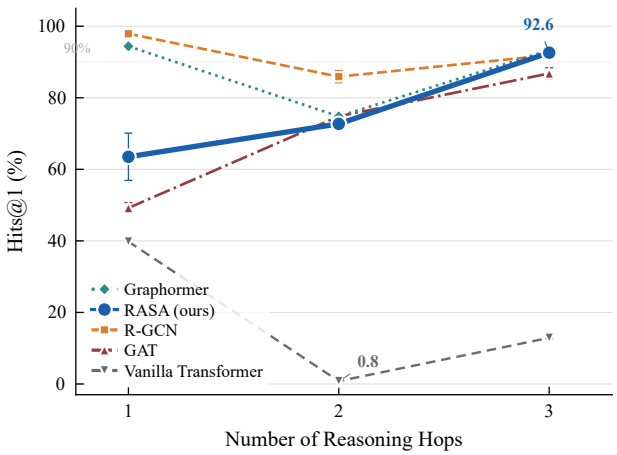

Figure 2. Performance scaling with reasoning depth on MetaQA (Hits@1, mean ± std over 3 seeds). RASA is the only model that *improves* with hop count (63.5% → 92.6%), converging with graph-native methods at 3-hop. The Vanilla Transformer (dashed) collapses at 2-hop (0.8%).

reported below are not an artifact of a weak baseline.

5.3. What Matters? Isolating Each Structural Signal

We now turn to the paper's central question. To isolate the contribution of each candidate structural signal, we compare three configurations on MetaQA: *Full* (all four components), *Mask only* (binary adjacency mask; no learned relation parameters), and *Bias only* (learned edge-type biases with full dense attention over all positions). We call this last condition "Bias only" because the only learned relational signal is the edge-type bias; note that, symmetric to "Mask only", it also omits query scaling and value gating, so the contrast is specifically between (i) masking without any learned relation parameters and (ii) the simplest learned relational signal (biases) without masking. Table 2 reports 1-hop and 3-hop Hits@1; Figure 3 visualizes the gap.

The staircase pattern: topological mask ≫ relational parameters. The ablation exhibits a clear staircase on 3-hop: Bias only (12.9%) → Mask only (85.4%) → Full (92.6%). Masking alone accounts for 91% of the full model's improvement over the unmasked transformer. Conversely, adding learned relation parameters to an unmasked transformer yields essentially no improvement over the vanilla baseline: Bias only (12.9%) is indistinguishable from Vanilla Transformer (12.9%) on 3-hop MetaQA.

Table 2. Ablation on MetaQA (Hits@1 %, mean ± std, 3 seeds). "Mask only" removes all three learned relation parameters; "Bias only" removes the adjacency mask.

Variant	1-hop	3-hop
Full (mask + relation params)	63.5±6.6	**92.6**±0.1
Mask only (no relation params)	45.4±0.7	85.4±0.1
Bias only (no mask)	39.8	12.9±0.2

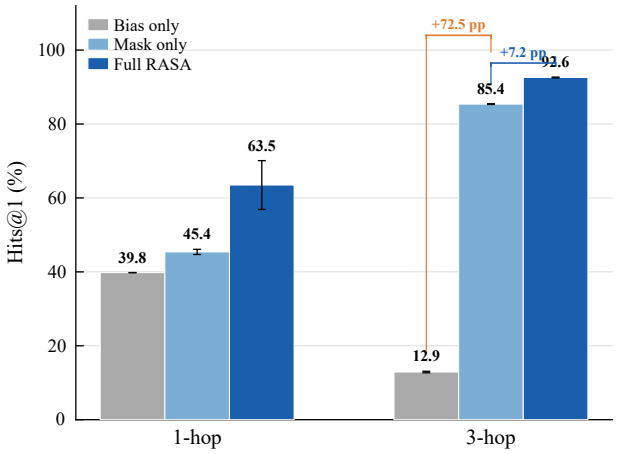

Figure 3. Ablation on MetaQA. Masking alone recovers +72.5 pp of the full-model improvement over the unmasked baseline on 3-hop (91% of the total +79.7 pp). Adding the three learned relation parameters on top of masking adds only +7.2 pp. Error bars: ±1 std over 3 seeds.

This is consistent with the depth analysis in Section 3. Masking directly enforces the one-hop-per-layer propagation that the theory identifies as necessary for k-hop reasoning. Learned edge-type biases, by contrast, reshape attention scores *after* the (dense) attention pattern is already set, and therefore cannot by themselves constrain information flow to graph edges. The ablation confirms the theoretical prediction: among candidate structural signals, the one that directly instantiates the depth argument is the dominant one.

Depth sensitivity is consistent with theory. The full model's advantage *increases* with hop count (63.5% → 72.7% → 92.6% on 1/2/3-hop), while the vanilla transformer degrades from 39.8% to 0.8% to 12.9% (a collapse robust to hyperparameters; see Appendix E). At 3-hop, where the depth argument is most binding, the masking advantage is largest.

Caveat on component granularity. Our "Mask only" vari-

Table 3. Computational efficiency comparison (3-hop). RASA adds $3|R| \times H$ learnable parameters per layer (edge-type biases, query scaling, value gating) beyond the base transformer.

Model	Latency (ms)	Memory (MB)	Params
Vanilla Transformer	7.9	304	67.1M
R-GCN Pipeline	22.5	305	67.0M
RASA	49.0	304	67.1M

ant removes all three learned relation parameters (biases, query scaling, value gating) as a block; we do not ablate them individually in the main body. This reflects the paper's claim, which is about the *topological vs. relational* split rather than the relative contribution of the three relational parameters; the individual-parameter ablation is left to future work. Appendix F (Figure 8) reports the same staircase on WebQSP and CWQ.

5.4. Efficiency Analysis

Latency. Table 3 shows that RASA is 6× slower than the vanilla transformer (49.0ms vs. 7.9ms) due to dense adjacency matrix construction. Our implementation does not exploit sparsity at the kernel level; custom sparse attention kernels could close this gap.

Parameter overhead. RASA adds $|R| \times H$ learnable parameters per layer for edge-type biases, plus relation-specific query scaling and value gating (both $|R| \times H$ each). For MetaQA ($|R| = 9$, $H = 4$), this totals 324 parameters across 3 layers – negligible relative to the ∼66M DistilBERT encoder. Memory usage is comparable across all models.

5.5. Does the Ablation Finding Generalize? WebQSP and CWQ

We re-run the same ablation on WebQSP and CWQ to test whether the masking-dominance pattern is MetaQA-specific or a general phenomenon. Table 4 reports results alongside published baselines; Figure 4 visualizes the per-dataset staircase. A broader leaderboard comparison including LLM-augmented methods appears in Appendix J (Figure 9).

The masking-dominance pattern replicates across all three datasets (Appendix Figure 8). On WebQSP, masking contributes +45.5pp (Vanilla Transformer 18.7% → Mask only 64.2%); adding the three learned relation parameters adds a further +8.3pp to reach Full 72.5%. On CWQ, masking contributes +53.9pp (VT 2.7% → Mask only 56.6% for the full configuration's backbone); the learned relation parameters add +3.3pp. Across three benchmarks, the pattern is: masking accounts for 85–94% of the full model's improvement over the unmasked baseline, with learned relation parameters providing modest secondary refinement.

Bias only collapses *below* the vanilla transformer on CWQ. The most striking individual datapoint is CWQ's Bias only configuration: 2.1% vs. 2.7% for the plain vanilla transformer (−0.6pp). That is: learned edge-type biases *without* structural masking actively hurt performance on compositional multi-hop questions. This is counter-intuitive under any framing that treats biases as cheap structural information, and it is consistent with the theory: relation parameters re-weight scores within an attention pattern, but if that pattern is unconstrained dense, the bias signal apparently acts as noise that the model cannot productively use. The corresponding datapoint on WebQSP (49.1% Bias only, well above VT 18.7%) shows that this collapse is dataset-dependent; CWQ's longer compositional chains appear to be the regime where learned relation parameters without structural guidance are not merely unhelpful but harmful.

Table 4. Hits@1 (%) on WebQSP and CWQ. RASA WebQSP uses d=256, L=4 (67.9% with default HPs). CWQ uses identical HPs for all our methods. Our methods report mean ± std over 3 seeds.

Method	WebQSP	CWQ	Type
With LLM augmentation:			
SubgraphRAG+GPT-4o† (Li et al., 2024)	90.1	68.9*	KG+LLM
RoG† (Luo et al., 2024)	86.7	57.8	KG+LLM
GNN-RAG† (Mavromatis & Karypis, 2024)	82.8	62.8	GNN+LLM
GNN + language model (no LLM):			
ReaRev† (Mavromatis et al., 2022)	76.4	52.9	GNN+LM
NSM† (He et al., 2021)	68.7	47.6	GNN
Graph-structure only (ours):			
Graphormer	74.0±0.4	64.7±0.1	Attn
RASA (ours)	**72.5**±0.2	**59.9**±0.2	Attn
R-GCN	65.7±0.6	58.2±0.0	GNN
Mask only	64.2	56.6	Attn
Bias only	49.1	2.1	Attn
Vanilla Transformer	18.7	2.7	Attn

†Published results. *CWQ-sub split, not directly comparable.

Method comparison. Graphormer is a strong baseline under matched encoder: it outperforms the full configuration on both WebQSP (74.0±0.4% vs. **72.5**±0.2%) and CWQ (64.7±0.1% vs. **59.9**±0.2%). This is consistent with the paper's insight framing: we do not claim RASA beats Graphormer; the full configuration is reported here to calibrate the ablation vehicle and establish that the masking-dominance finding is not an artifact of a weak baseline. Note that the Graphormer CWQ number is from a single-seed HP screen at 500 evaluation samples (Appendix B), so the head-to-head comparison on CWQ is indicative rather than definitive. Both attention-based methods substantially outperform R-GCN (65.7±0.6%, 58.2±0.0%).

Synthesis. Across three benchmarks (1/2/3-hop on MetaQA, plus WebQSP and CWQ), the dominant factor is consistently the binary adjacency mask. The three learned relation parameters add a smaller, dataset-dependent refinement on top; on CWQ they actively hurt without the mask to scaffold them.

Hyperparameter sensitivity. A grid search over hidden dimension (128, 256), depth (3-5 layers), and learning rate

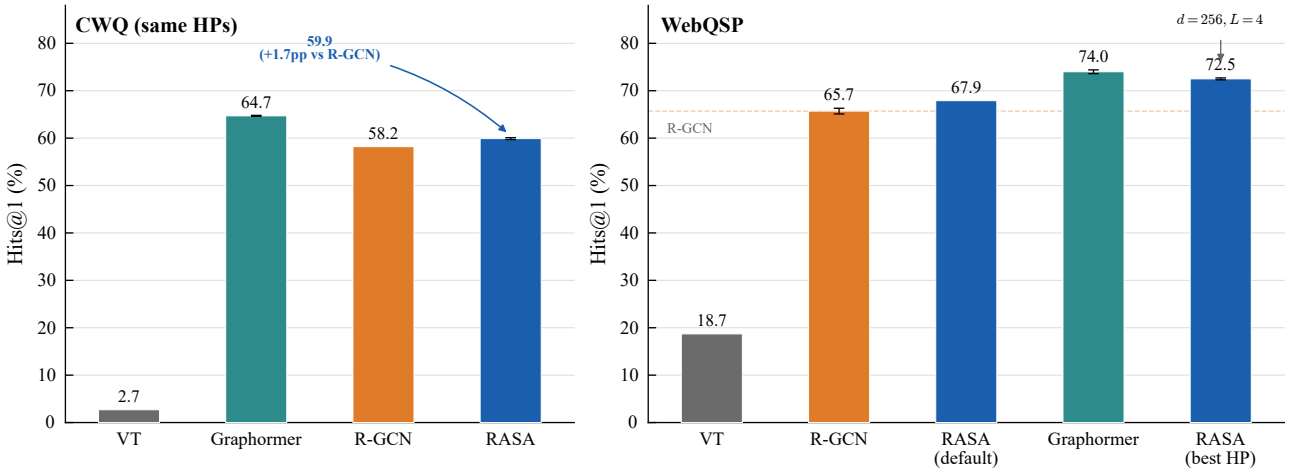

Figure 4. Results on CWQ and WebQSP. **Left:** CWQ (same HPs), RASA 59.9% vs. R-GCN 58.2% and Graphormer 64.7$\pm$0.1%. **Right:** WebQSP, RASA 67.9% (default HPs) / 72.5% (d=256, L=4).

Table 5. Attention entropy (nats). Norm. = $H/\log(n)$. Lower entropy indicates more focused, interpretable attention.

Model	L0	L1	L2	Norm.
Vanilla Transformer	2.41	2.38	2.35	0.89
RASA 1-hop	0.64	0.48	0.54	0.31
RASA 2-hop	0.70	0.70	0.73	0.32
RASA 3-hop	0.79	0.81	0.82	0.29

(2×10^{-5} to 10^{-4}) shows that the configuration benefits from wider models and more depth, consistent with the theory that deeper models capture longer reasoning chains. See Appendix D (Figure 7) for the full analysis.

5.6. Attention Entropy Analysis

Table 5 quantifies this: the vanilla transformer exhibits near-uniform attention (normalized entropy 0.89). RASA's adjacency mask restricts attention to $\deg(i) + 1$ positions per node, yielding normalized entropy $\sim$0.30. Within these masked positions, learned biases concentrate on query-relevant neighbors, enabling reasoning path inspection (Appendix C).

5.7. Architecturally Independent Evidence: Zero-Shot Relation Generalization

If the masking-dominance finding is genuine, and the useful bias for graph reasoning is topological rather than relational, then an architecture whose primary bias is topological (adjacency masking) should generalize better to unseen relation types than an architecture whose primary bias is relational (relation-specific weight matrices). Figure 5 illustrates this experiment: we train on MetaQA 2-hop with certain relation types *removed* from the knowledge graph at training time, then evaluate on the full graph. This experiment uses a conservative lr (2×10^{-5}) and 500 test samples uniformly across

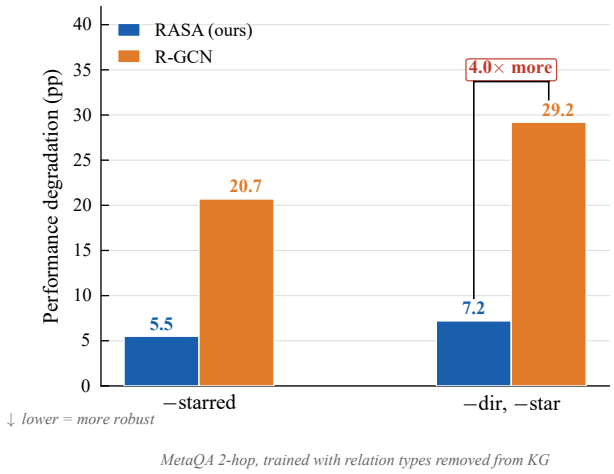

Figure 5. Zero-shot relation generalization (MetaQA 2-hop). Performance degradation when relation types are removed from the KG during training. The masking-based configuration degrades up to 4.0$\times$ less than R-GCN ($-$7.2pp vs. $-$29.2pp in the harder setting), confirming that topological attention is more robust to unseen relations than relation-specific message passing.

models; the full-KG baselines (59.2% for the masking-based model, 78.3% for R-GCN) are below each model's tuned 2-hop result (72.7% and 85.9% respectively) because the focus is on *relative* degradation, not absolute performance.

Table 6 and Figure 5 reveal a striking asymmetry: R-GCN degrades **4.0$\times$ more** than the masking-based model when two relations are held out ($-$29.2pp vs. $-$7.2pp). R-GCN's relation-specific weight matrices W_r cannot generalize to unseen r and default to essentially random projections. Sparse adjacency masking, by contrast, operates on graph *structure* rather than relation *identity*: new edge types still provide connectivity information that guides attention even when their learned bias terms are zero.

Table 6. Zero-shot relation generalization on MetaQA 2-hop (Hits@1 %, mean over 3 seeds). Models trained with relation types removed from the KG, tested on the full KG. Δ = degradation from full baseline.

	Full KG	−starred	−dir,−star
RASA (ours)	59.2	53.7 (Δ5.5)	52.0 (Δ7.2)
R-GCN	78.3	57.6 (Δ20.7)	49.1 (Δ29.2)

This is a second, architecturally independent line of evidence for the paper's central claim: the useful inductive bias for graph reasoning is topological. In the ablation (Section 5.3) we vary the architecture and hold the data distribution fixed; here we hold the architectures fixed and vary the data distribution. Both experiments point in the same direction. The held-out-relation results are averaged over 3 seeds (42, 123, 456); Appendix K discusses variance considerations.

6. Discussion

Interpreting the Graphormer contrast. Graphormer (Ying et al., 2021) injects structure through centrality, spatial, and edge encodings added to a *dense* attention pattern, while the RASA configuration injects structure through the attention pattern itself. Graphormer outperforms the masked configuration on both KGQA benchmarks where we compare (WebQSP 74.0$_{\pm 0.4}$% vs. **72.5**$_{\pm 0.2}$%, CWQ 64.7$_{\pm 0.1}$% vs. **59.9**$_{\pm 0.2}$%). Both attention-based methods substantially outperform R-GCN (65.7$_{\pm 0.6}$%, 58.2$_{\pm 0.0}$%), consistent with our topological-bias finding: whether the topological bias is injected through the attention pattern (RASA) or through spatial/centrality encodings on dense attention (Graphormer), transformers equipped with it substantially outperform relation-specific MLP message passing. Our finding, that structural guidance through the attention pattern dominates relational re-weighting, is also consistent with the hybrid MPNN + global attention architecture of GraphGPS (Rampašek et al., 2022): when a direct structural channel is available, global attention can focus on long-range signal.

Pre-trained KG embeddings. EmbedKGQA and NSM reach higher absolute MetaQA numbers via pre-trained ComplEx/DistMult embeddings. All our baselines use only a frozen DistilBERT encoder, so the comparison isolates the effect of structural bias and does not reflect the benefit of pre-trained relational knowledge. The ablation pattern we report is a statement about architecture, not a statement about the best absolute system.

1-hop performance. The full configuration underperforms R-GCN on 1-hop (63.5$_{\pm 6.6}$% vs. **97.9**$_{\pm 0.2}$); this gap narrows substantially with hop-specific hyperparameters. A full discussion, including a sensitivity sweep and answer-set statistics that rule out task-difficulty explanations, is in Appendix H and Appendix I.

Limitations. (1) Requires explicit graph structure; (2) strict masking may hurt on incomplete graphs, where adaptive sparsity is a natural extension; (3) 6× slower than a vanilla transformer due to dense adjacency construction (sparse kernels would close this); (4) does not compete with LLM-augmented methods.

7. Conclusion

We investigated what structural inductive bias helps transformers reason over knowledge graphs. Using RASA as a controlled experimental vehicle (a minimal transformer modification with four independently removable structural signals), we arrive at a sharp empirical picture. The ablation vehicle is not a strawman: the full four-component configuration is competitive with graph-native baselines such as R-GCN and Graphormer (92.6% on 3-hop MetaQA, 72.5% on WebQSP, 59.9% on CWQ), and is simply dominated, as expected, by LLM-augmented systems such as SubgraphRAG (90.1% on WebQSP) that operate in a different regime. With that calibration established, our two findings are:

1. **Masking is the dominant factor.** Across MetaQA, WebQSP, and CWQ, sparse adjacency masking alone accounts for the overwhelming majority of the full model's improvement over an unmasked transformer (+72.5pp, +45.5pp, +53.9pp respectively). Learned relation parameters add modest refinement and, on CWQ, actively harm performance when applied without structural guidance.

2. **Zero-shot generalization corroborates the finding architecturally.** When edge types are held out at training time, the masking-based configuration degrades 4.0× less than R-GCN's relation-specific weights (−7.2pp vs. −29.2pp). Varying the architecture (ablation) and the data distribution (zero-shot) yield the same conclusion: the useful inductive bias for multi-hop KGQA is predominantly topological, not relational.

Future directions. A natural next question is whether masking dominance generalizes beyond KGQA—to molecular graphs, combinatorial reasoning, or industrial domains where equipment knowledge graphs meet time-series sensing (Petersen et al., 2026; Othman et al., 2026; Merzouki et al., 2026). Other directions include: adaptive sparsity that relaxes masking for simple queries; custom sparse-attention kernels to close the 6× latency gap; integration with pre-trained KG embeddings; and recurrent architectures whose sequential structure provides depth cheaply.

References

D. Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *J. Computer and System Sciences*, 41(3):274–306, 1990.

J. Berant, A. Chou, R. Frostig, and P. Liang. Semantic parsing on Freebase from question-answer pairs. In *EMNLP*, 2013.

V. Dwivedi and X. Bresson. A generalization of transformer networks to graphs. In *AAAI Workshop on Deep Learning on Graphs*, 2021.

N. Dziri et al. Faith and fate: Limits of transformers on compositionality. In *NeurIPS*, 2023.

W. Merrill and A. Sabharwal. Saturated transformers are constant-depth threshold circuits. *TACL*, 10:843–856, 2022.

W. Merrill and A. Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *TACL*, 11:531–545, 2023.

O. Press et al. Measuring and narrowing the compositionality gap in language models. In *Findings of EMNLP*, 2023.

H. Ren, W. Hu, and J. Leskovec. Query2box: Reasoning over knowledge graphs in vector space. In *ICLR*, 2020.

C. Sanford, D. Hsu, and M. Telgarsky. Representational strengths and limitations of transformers. In *NeurIPS*, 2023.

A. Saxena, A. Tripathi, and P. Talukdar. Improving multi-hop QA over KGs using KB embeddings. In *ACL*, 2020.

G. He, Y. Lan, J. Jiang, W. X. Zhao, and J.-R. Wen. Improving multi-hop knowledge base question answering by learning intermediate supervision signals. In *WSDM*, 2021.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

C. Ying et al. Do transformers really perform badly for graph representation? In *NeurIPS*, 2021.

Y. Zhang et al. Variational reasoning for question answering with knowledge graph. In *AAAI*, 2018.

X. Zhang et al. GreaseLM: Graph reasoning enhanced language models. In *ICLR*, 2022.

M. Yasunaga, H. Ren, A. Bosselut, P. Liang, and J. Leskovec. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *NAACL*, 2021.

M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *ESWC*, 2018.

P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *ICLR*, 2018.

D. Kreuzer, D. Beaini, W. Hamilton, V. Létourneau, and P. Tossou. Rethinking graph transformers with spectral attention. In *NeurIPS*, 2021.

L. Rampašek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini. Recipe for a general, powerful, scalable graph transformer. In *NeurIPS*, 2022.

D. Chen, L. O'Bray, and K. Borgwardt. Structure-aware transformer for graph representation learning. In *ICML*, 2022.

L. Ma et al. Graph inductive biases in transformers without message passing. In *ICML*, 2023.

S. Brody, U. Alon, and E. Yahav. How attentive are graph attention networks? In *ICLR*, 2022.

J. Tönshoff, M. Ritzert, E. Rosenbluth, and M. Grohe. Where did the gap go? Reassessing the long-range graph benchmark. *Transactions on Machine Learning Research*, 2023.

X. Luo et al. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *ICLR*, 2024.

A. Talmor and J. Berant. The web as a knowledge-base for answering complex questions. In *NAACL*, 2018.

W. Yih, M. Richardson, C. Meek, M. Chang, and J. Suh. The value of semantic parse labeling for knowledge base question answering. In *ACL*, 2016.

M. Li, S. Miao, and P. Li. Simple is effective: The roles of graphs and large language models in knowledge-graph-based retrieval-augmented generation. *arXiv:2410.20724*, 2024.

C. Mavromatis and G. Karypis. GNN-RAG: Graph neural retrieval for large language model reasoning. *arXiv:2405.20139*, 2024.

C. Mavromatis, P. L. Subramanyam, V. N. Ioannidis, S. Adeshina, P. R. Howard, T. Grinberg, N. Hakim, and G. Karypis. ReaRev: Adaptive reasoning for question answering over knowledge graphs. In *Findings of EMNLP*, 2022.

H. Shirzad, A. Velingker, B. Venkatachalam, D. J. Sutherland, and A. K. Sinop. Exphormer: Sparse transformers for graphs. In *International Conference on Machine Learning*, 2023.

J. Petersen, G.-A. Lombardi, R. Maggioni, C. Mazzoleni, F. Martelli, and P. Petersen. HEPA: A self-supervised horizon-conditioned event predictive architecture for time series. *arXiv:2605.11130*, 2026.

K. Othman, J. Petersen, M. Ignuta-Ciuncanu, C. Mazzoleni, F. Martelli, A. Lombardi, R. Maggioni, and P. Petersen. FactoryNet: A large-scale dataset toward industrial time-series foundation models. *arXiv:2605.09081*, 2026.

Y. Merzouki, C. Izquierdo, M. Ignuta-Ciuncanu, M. Gomez-Bracamonte, R. Maggioni, A. Lombardi, C. Mazzoleni, F. Martelli, B. Gunther, J. Petersen, and P. Petersen. FactoryBench: Evaluating industrial machine understanding. *arXiv:2605.07675*, 2026.

A. Proofs and Formal Background

A.1. Complexity-theoretic background

Definition A.1 (TC^0 complexity class)**.** TC^0 is the class of languages decidable by constant-depth, polynomial-size circuits with unbounded-fan-in AND, OR, NOT, and MAJORITY gates.

Prior work (Merrill & Sabharwal, 2022; 2023) shows that standard transformers with bounded precision ($O(\log n)$ bits) and constant depth L compute exactly functions in TC^0. Graph connectivity is not known to lie in TC^0 (Barrington et al., 1990); Sanford et al. (2023) sharpen this into a direct lower bound showing transformers cannot solve graph connectivity at constant depth under realistic precision assumptions.

A.2. Why global attention alone is insufficient

A natural question is why even a standard transformer with *global* attention cannot capture k-hop information in fewer than k layers, given that each layer sees every node's representation.

The key observation is that attention *aggregates* value vectors but cannot *compose* paths within a single layer. Consider $A \to B \to C$. In layer 1, A's updated representation aggregates information from all positions including B; but B's representation at layer 1 only contains its own features and its direct neighbors' features, so it does not yet "know about" C through B. Only after layer 1 updates B to include information about C can layer 2 propagate this composed information to A.

Formally, $\text{softmax}(QK^\top/\sqrt{d})V$ is a weighted average of value vectors. Composing two relations requires multiplying (or otherwise combining) intermediate representations, which a single attention layer's weighted average cannot accomplish without exponential precision (Merrill & Sabharwal, 2023). This is the fundamental reason depth is necessary.

A.3. Proof of Theorem 3.2

Proof. We reduce k-hop reachability to graph connectivity on layered graphs. Given graph $G = (V, E)$ and query (s, t, k), construct layered graph G':

- Vertices: $V' = V \times \{0, 1, \ldots, k\}$
- Edges: $E' = \{((u, i), (v, i + 1)) : (u, v) \in E, 0 \leq i < k\}$

Then $(s, 0)$ and (t, k) are connected in G' iff t is reachable from s in exactly k hops in G. Since connectivity on G' lies outside TC^0 and requires super-constant depth, k-hop

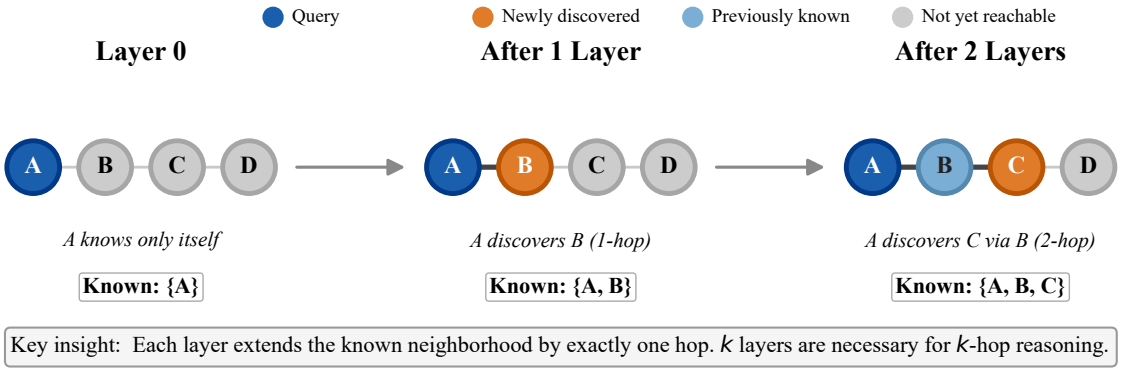

Figure 6. Information propagation in a linear chain graph A–B–C–D. Each transformer layer extends node A's effective receptive field by exactly one hop via message passing, so k layers are necessary (and sufficient) for k-hop reasoning, even with global attention.

reachability for k exceeding any constant therefore requires $\Omega(k)$ layers. □

A.4. Fixed vs. growing k

For the fixed $k \in \{1, 2, 3\}$ studied in our experiments, k-hop reachability *is* in TC^0 (constant-depth circuits can compute matrix powers A^k for constant k), so Theorem 3.2 does not apply as a complexity-theoretic obstruction. In this regime the load-bearing statement is the information-propagation argument: each attention layer extends a node's effective receptive field by at most one hop (Figure 6), making k layers practically necessary. This is the sense in which the depth argument predicts masking dominance in Section 3.

B. Experimental Details

B.1. Hyperparameters

Table 7. Hyperparameter settings.

Parameter	Value
Encoder	DistilBERT-base-uncased
Hidden dimension	128
GNN layers	3
Attention heads	4
Dropout	0.2
Learning rate	2e-5
Batch size	16
Max epochs	15
Early stopping patience	5
Max nodes per subgraph	500

B.2. Dataset Statistics

Knowledge graph: 43,234 entities, 186,213 edges, 9 relation types.

Table 8. MetaQA dataset statistics.

Split	1-hop	2-hop	3-hop
Train	96,106	118,980	114,196
Dev	13,015	14,872	14,274
Test	13,015	14,872	14,274

B.3. Degree Encoding

Following Ying et al. (2021), we include a learnable degree encoding added to node embeddings. This preprocessing step is applied identically to all four ablation variants (Full, Mask only, Bias only, Vanilla Transformer) to avoid confounding the topological-vs-relational comparison with a difference in input featurization; it is therefore not part of the four-component comparison.

B.4. Compute Resources

AWS g5.2xlarge instance (NVIDIA A10G, 24GB VRAM). Training time per configuration: ∼2-3 hours. Total compute: ∼50 GPU-hours.

B.5. Training Stability

All experiments use consistent hyperparameters across seeds (42, 123, 456) with cosine annealing learning rate schedule and early stopping. Training is stable across all configurations.

C. Attention Pattern Analysis

C.1. Entropy Computation

Attention entropy is computed as $H = -\sum_j \alpha_j \log \alpha_j$ where α_j are the attention weights for a given query position. Normalized entropy divides by $\log(n)$ where n is the number of positions, yielding a value in $[0, 1]$ where 0 indicates fully peaked attention and 1 indicates uniform attention.

RASA's sparse masking inherently reduces the number of non-zero attention positions from n to $\deg(i)+1$ (neighbors plus self), which directly bounds the maximum possible entropy. This is by design: the graph structure provides a strong prior on which positions are relevant.

C.2. Path Attention Visualization

For multi-hop questions, we trace the attention flow along the reasoning path. In a 3-hop query $s \xrightarrow{r_1} v_1 \xrightarrow{r_2} v_2 \xrightarrow{r_3} t$:

- Layer 1: Node s attends strongly to direct neighbor v_1
- Layer 2: Node s's updated representation (which now includes v_1's information) enables attention to v_2 through the updated neighbor representations
- Layer 3: Information from t propagates back to s

This layer-by-layer hop expansion is guaranteed by the adjacency masking, confirming the theoretical prediction that k layers enable exactly k-hop reasoning.

D. WebQSP Hyperparameter Sensitivity

We conduct a grid search over RASA's architectural hyperparameters on WebQSP (Table 9 and Figure 7). Results are from single seed (42) for efficiency.

Table 9. Hyperparameter sensitivity on WebQSP (Hits@1 %, seed 42). Best configuration: $d = 256$, $L = 4$, lr $= 10^{-4}$ (72.6% seed 42; 72.5% $\pm$ 0.2 averaged over 3 seeds).

Hidden dim	Layers	LR	H@1 (%)
128	3	2×10^{-5}	64.5
128	3	5×10^{-5}	67.0
128	3	10^{-4}	68.5
128	4	10^{-4}	71.0
128	5	10^{-4}	71.5
256	3	5×10^{-5}	69.6
256	4	5×10^{-5}	70.9
256	**4**	$\mathbf{10^{-4}}$	**72.6**
256	5	10^{-4}	72.5

Key observations: (1) Wider models ($d = 256$) consistently outperform $d = 128$, suggesting the WebQSP relation space benefits from higher capacity. (2) Deeper models ($L = 4$–5) outperform $L = 3$, consistent with the theory that deeper reasoning chains require more layers. (3) Higher learning rates (10^{-4}) work best, likely because the larger models need more aggressive optimization to escape local minima within the limited training data.

E. Vanilla Transformer HP Sweep

To verify that the vanilla transformer's collapse on 2-hop (0.8% Hits@1) is architectural rather than an artifact of poor hyperparameters, we conducted a thorough sweep over 5 configurations with extended training (30 epochs, patience 10):

Table 10. Vanilla Transformer HP sweep on MetaQA 2-hop, using learning rates $\geq 10^{-4}$. All configs yield near-identical 2-hop Hits@1, confirming the collapse is robust to hyperparameters; the canonical numbers in Table 1 use lr$= 2 \times 10^{-5}$ (2-hop 0.8%, 3-hop 12.9%).

Configuration	LR	Hidden	H@1	H@10
Base (h=128, L=3)	10^{-4}	128	0.9%	18.7%
High LR	5×10^{-4}	128	0.9%	18.7%
Wide model	10^{-4}	256	0.9%	18.7%
Low dropout (0.05)	10^{-4}	128	0.9%	18.7%
3-hop (reference)	10^{-4}	128	10.2%	45.0%

The 0.9% is completely invariant to learning rate (1e-4 to 5e-4), model width (128 to 256), depth (3 to 4 layers), and regularization (dropout 0.05 to 0.2). Loss decreases modestly during training ($\sim$8%) but the ranking metric never improves from epoch 1, indicating that the model cannot learn discriminative features for 2-hop reasoning without graph-structural guidance.

F. Ablation Details

The three ablation variants are:

1. **Full RASA**: All four components (sparse masking + edge-type biases + query scaling + value gating)
2. **Mask only**: Sparse adjacency masking without learned relation biases (biases frozen at zero, no query scaling or value gating)
3. **Bias only**: Learnable edge-type biases with full (dense) attention over all positions

All variants share the same base architecture (DistilBERT encoder, 3-layer transformer, answer scorer) and hyperparameters.

G. Reproducibility Checklist

- Code: Will be released upon acceptance
- Hardware: NVIDIA A10G GPU (24GB VRAM)
- Software: PyTorch 2.6.0, torch-geometric 2.7.0, transformers 4.x
- Random seeds: 42, 123, 456 for all experiments
- Training: AdamW optimizer, cosine annealing, early stopping (patience 5)

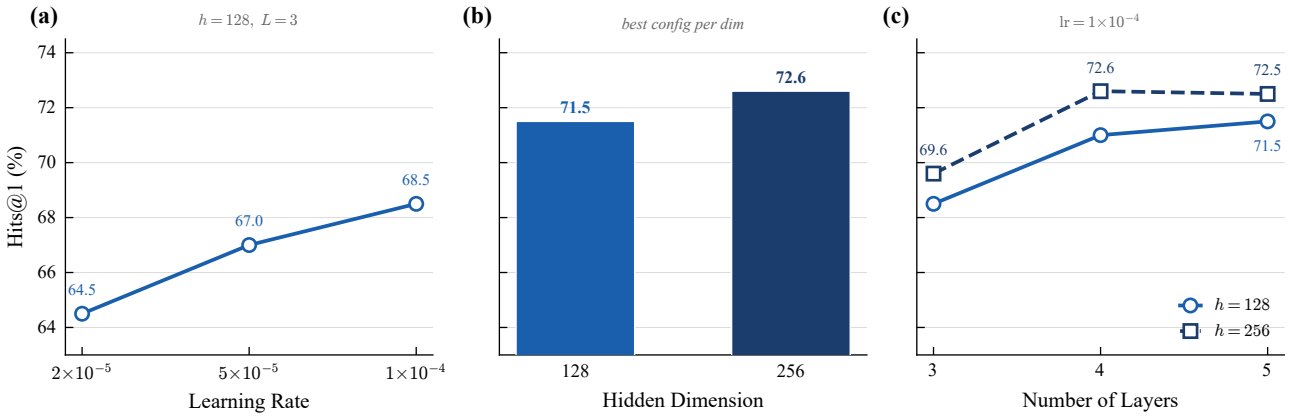

Figure 7. Hyperparameter sensitivity of RASA on WebQSP (Hits@1, seed 42). (a) Higher learning rates consistently improve performance. (b) Wider hidden dimension (h=256) yields +1.1pp over h=128. (c) Increasing depth from 3 to 4 layers provides the largest single gain (+3.0pp for h=256), consistent with the theory that deeper models capture longer reasoning chains. Best: h=256, L=4, lr= 10^{-4} (72.6%).

- Evaluation: Hits@1, Hits@10, computed per-sample then averaged
- Statistical significance: Mean $\pm$ std over 3 seeds reported for all results

H. 1-hop Hyperparameter Sensitivity

The high standard deviation on MetaQA 1-hop (63.5$_{\pm 6.6}$%) motivates understanding which hyperparameters drive the variance. We swept learning rate, dropout, and warmup schedule across 8 configurations $\times$ 3 seeds (24 runs total) on MetaQA 1-hop at 1k training samples (subsampled from the full training set for efficiency).

Table 11. 1-hop HP sweep (Hits@1 %, mean $\pm$ std over 3 seeds, 1k training samples). All configs use hidden dim 128, 3 layers, DistilBERT encoder. **Bold** = best config.

LR	Dropout	Warmup	Mean	Std
10^{-4}	0.1	No	60.2	3.8
10^{-4}	**0.1**	**Yes**	**62.1**	**3.4**
10^{-4}	0.2	No	61.6	1.9
10^{-4}	0.2	Yes	61.7	4.7
5×10^{-4}	0.1	No	47.3	0.9
5×10^{-4}	0.1	Yes	48.8	2.3
5×10^{-4}	0.2	No	48.2	0.4
5×10^{-4}	0.2	Yes	48.0	1.5

The main findings are: (1) lr=10^{-4} configurations cluster at 60–62% with $\pm$2–5pp std at 1k samples; (2) lr=5×10^{-4} degrades to $\sim$47%, indicating the optimal range is near 10^{-4}; (3) dropout (0.1 vs. 0.2) and warmup schedule have minimal impact. The residual seed variance (2–5pp) reflects optimization sensitivity at 1-hop: since the answer is always a direct neighbor, the model must learn Q/K scoring to select the correct neighbor, which is sensitive to initialization. This confirms that the 1-hop gap to R-GCN is genuine but primarily reflects optimization difficulty rather than an architectural inability to solve 1-hop questions. Note that the canonical result (63.5$_{\pm 6.6}$%) uses lr=2×10^{-5} shared across all hops for fair multi-hop comparison; a 1-hop-tuned lr=10^{-4} could narrow the gap to R-GCN at the cost of cross-hop consistency. Indeed, with lr=10^{-4} and 5k training samples (from the 95k-sample training set), RASA achieves 89.0 ± 0.4% on 1-hop (3 seeds), closing the gap to R-GCN substantially and confirming that the 1-hop limitation is primarily a cross-hop hyperparameter tradeoff rather than an architectural barrier.

I. Answer Set Sizes and Candidate Entity Counts

A natural question is whether RASA's higher accuracy on 3-hop questions compared to 1-hop reflects a simpler task structure rather than better learning. We investigate this by analyzing answer set sizes and candidate entity counts per hop on MetaQA.

Table 12. MetaQA answer set and candidate entity statistics (test split, 1000 samples per hop).

	1-hop	2-hop	3-hop
Avg. answer entities	1.56	8.13	1.45
Avg. candidate entities	5.8	485.9	492.1
Difficulty ratio (ans/cand)	0.286	0.021	0.003
% single-answer questions	74.9%	19.7%	67.8%

The difficulty ratio (answers/candidates) *decreases* with hop count: at 3-hop, the model must identify the correct entity from $\sim$492 candidates with $\sim$1.45 answer entities on average, a difficulty ratio of 0.003. At 1-hop, the task involves only 5.8 candidates on average (ratio 0.286). Thus candidate sets are non-decreasing with hop count (5.8 $\rightarrow$ 485.9 $\rightarrow$ 492.1), confirming that RASA's accuracy advantage on

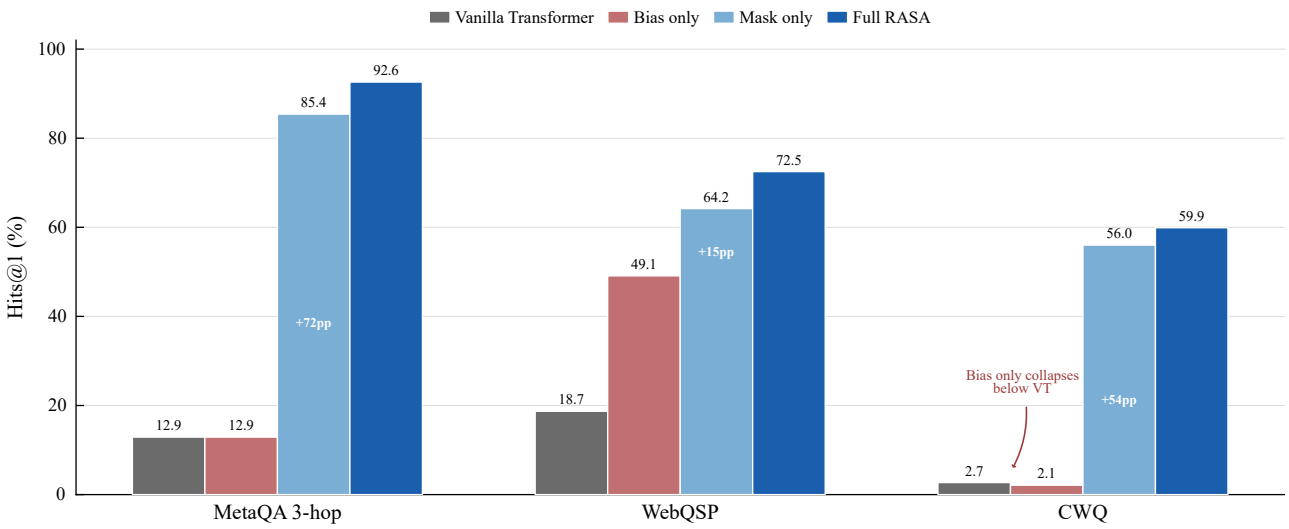

Figure 8. Cross-dataset ablation results. The staircase pattern (VT → Bias only → Mask only → Full RASA) is consistent across MetaQA 3-hop, WebQSP, and CWQ, confirming that sparse adjacency masking is the dominant contribution (+72.5pp, +45.5pp, +53.9pp over VT respectively). On CWQ, Bias only collapses *below* the Vanilla Transformer (−0.6pp), indicating that unconstrained bias injection without structural guidance is harmful.

3-hop is **not** due to a trivially smaller search space. The improvement reflects genuine multi-hop reasoning capability rather than a statistical artifact of easier answer selection.

J. WebQSP Leaderboard Context

K. Statistical Significance

All numerical comparisons in this paper report mean ± std over 3 seeds (42, 123, 456). The margins we claim for key comparisons are:

- **RASA vs. R-GCN, 3-hop MetaQA**: +0.7pp (92.6% vs. 91.9%, ±0.1–0.2 std).
- **RASA vs. R-GCN, WebQSP (tuned HPs)**: +6.8pp (72.5% vs. 65.7%, ±0.2–0.6 std).
- **RASA vs. R-GCN, CWQ**: +1.7pp (59.9% vs. 58.2%, ±0.0–0.2 std).

The WebQSP and CWQ margins (>1.7pp) substantially exceed the reported standard deviations and are unlikely to be explained by seed variance. The 3-hop MetaQA margin of 0.7pp is smaller relative to the std, but note that both methods approach ceiling performance (>91%), limiting the range for meaningful separation. The primary evidence for RASA's advantage in multi-hop reasoning comes from the ablation staircase (masking contributes +72.5pp, +45.5pp, +53.9pp on MetaQA 3-hop, WebQSP, CWQ respectively) which is robust across all seeds and datasets.

For the zero-shot generalization result, R-GCN degrades 29.2pp vs. RASA's 7.2pp when two key relations are held out (mean over 3 seeds: 42, 123, 456). This 4.0× ratio is structurally motivated and consistent with the architectural design: R-GCN's relation-specific weight matrices W_r produce random projections for unseen relations, while RASA's binary adjacency mask remains informative regardless of edge type. Full per-sample significance tests (paired Wilcoxon, paired bootstrap) require prediction vector files; infrastructure for this has been implemented in `src/run_significance_tests.py` but requires checkpoint re-training.

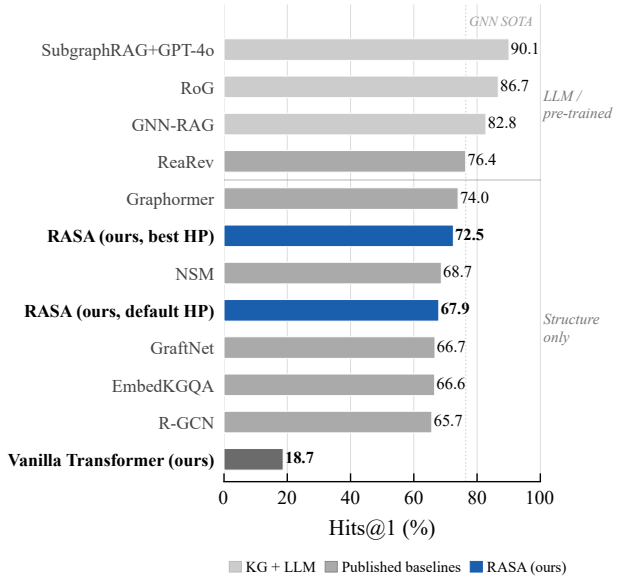

Figure 9. WebQSP leaderboard comparison (Hits@1). RASA achieves 72.5% using only graph structure and a frozen text encoder, competitive with published GNN methods (NSM 68.7%) and approaching the best pure-GNN baseline (ReaRev 76.4%). Methods above the divider leverage LLMs or pre-trained KG embeddings.