HKVM: LEARNING TO SELECT MULTI-HOP EVIDENCE WITH HYPERGRAPH KEY-VALUE MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-hop retrieval-augmented generation depends on selecting a small set of passages that jointly support an answer within a limited context budget. Relational indexes expose connections across passages, but those connections do not by themselves determine which evidence should be retained. We introduce HKVM (Hypergraph Key-Value Memory), a framework that couples relational indexing with learned evidence selection. HKVM groups locally extracted triples around shared bridge entities to form hypergraph retrieval keys, while retaining the original passages as values. A supervised selector combines structural and text-retrieval signals to rank candidate passages. We evaluate HKVM on the development splits of 2WikiMultiHopQA, MuSiQue, and HotpotQA with fixed candidate pools and an up-to-five-passage selection budget. Compared with a learned controller built on the same ColBERTv2 retrieval source, HKVM improves support-selection F1 (precision-recall F1 over supporting passages) by 1.61, 1.52, and 0.36 points and increases the fraction of questions retaining all supporting passages by 6.54, 3.57, and 1.16 percentage points, respectively. Across six retrieval sources, supervised reranking improves support-selection F1 for every source, while comparisons among learned alternatives vary by dataset and by whether performance is measured by average support or complete-support retention. These results show that relational indexing is most useful as a learned evidence-control layer: its value lies in how structural signals guide passage selection and preserve complete multi-hop support.

1 INTRODUCTION

Multi-hop question answering (Yang et al., 2018; Ho et al., 2020; Trivedi et al., 2022) often depends on facts distributed across several passages. In retrieval-augmented generation (Lewis et al., 2020), however, only a small set can fit within the context budget. The retrieval problem is therefore not only to find relevant passages, but to retain a set whose members jointly support the question.

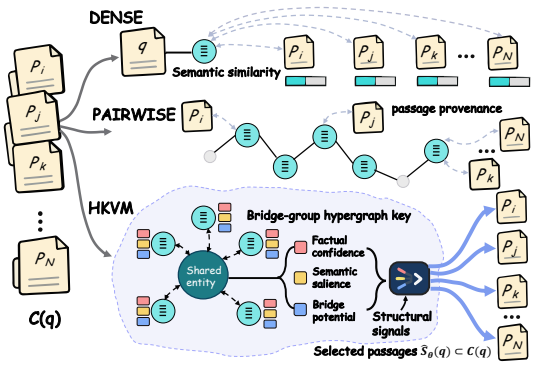

Figure 1: **HKVM address spaces.** Dense, pairwise, and HKVM organize the same candidate passages differently; HKVM uses bridge-group keys for structural signals while original passages remain the selected values.

Dense retrievers (Karpukhin et al., 2020; Izacard et al., 2022; Santhanam et al., 2022) score passages largely independently, whereas graph-based systems (Edge et al., 2024; Gutiérrez et al., 2024) expose relations that connect evidence across passages. As Figure 1 makes concrete, these are different ways of addressing the same candidate pool. Relational visibility alone, however, does not determine which connected passages should be selected: a shared entity may link complementary facts, redundant passages, or material that does not support the current question.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Other multi-hop systems alter the retrieval trajectory itself, for example through condensed retrieval, interleaved reasoning, adaptive retrieval, or retrieval-aware generation (Khattab et al., 2021; Trivedi et al., 2023; Jeong et al., 2024; Asai et al., 2024). These strategies address how a system searches or controls retrieval; they leave open how relational organization should inform the final passage set. Relational organization and evidence selection are thus distinct parts of the problem.

We study this distinction through HKVM (Hypergraph Key-Value Memory). It groups locally extracted relation tuples around shared bridge entities to form hypergraph keys that address sets of passages. The original passages remain the readable content values, while a supervised passage-level selector combines structural signals with text-retrieval features to rank candidates under the available context budget.

We evaluate this interface on the development splits of three multi-hop benchmarks using fixed candidate pools and an available-prefix budget of up to five passages. Across the eighteen source–dataset comparisons, passage-level supervision improves support-selection F1 for every retrieval source. Once sources receive the same learning opportunity, the relative utility of the native relational controller varies with the dataset, learned comparator, and support objective. Diagnostics further distinguish evidence missing from the candidate pool from evidence lost during selection, and show that relational groups can contain both support-related signal and distractors.

Contributions. (1) We formulate multi-hop retrieval as budgeted support selection and introduce HKVM, which separates relational passage addressing from the original passage values. (2) We establish a controlled comparison of raw and learned retrieval sources on fixed candidate pools, making the effect of passage-level supervision separable from the effect of relational organization. (3) We provide diagnostics for candidate visibility, group composition, and support objectives, showing when relational structure supplies useful selection signal and when its utility becomes conditional on the dataset, comparator, and target being optimized.

2 Related Work

Multi-hop retrieval and structured reasoning. Multi-hop retrieval methods change the search trajectory or interleave retrieval with reasoning. Baleen uses condensed retrieval (Khattab et al., 2021), IRCoT interleaves retrieval with chain-of-thought steps (Trivedi et al., 2023), Adaptive-RAG routes by question complexity (Jeong et al., 2024), and Self-RAG retrieves, generates, and critiques through reflection (Asai et al., 2024). Think-on-Graph and StructGPT instead reason over or interface with structured representations (Sun et al., 2024; Jiang et al., 2023). These works treat evidence access as a controlled process; HKVM asks how a relational address space can provide features to a passage-level selector.

Structured evidence retrieval. Graph-based retrieval exposes relations among entities and passages: GraphRAG organizes document communities (Edge et al., 2024), HippoRAG propagates query activation over a phrase graph (Gutiérrez et al., 2024), and HippoRAG 2 extends this with deeper passage integration and online LLM use (Gutiérrez et al., 2025). G-Retriever couples graph representations with language models for textual graph QA (He et al., 2024). Hyper-RAG and HyperGraphRAG represent higher-order relations with hypergraphs (Feng et al., 2026; Luo et al., 2026), while Cross-Granularity Hypergraph RAG treats passages as hyperedges over entity nodes (Wang et al., 2026). HKVM complements these designs by forming passage-address groups from tuples that share a bridge entity, then feeding their structural signals to a selector whose output remains the original passages.

Addressing, content, and selection. Hypergraph learning supports grouping beyond pairs through neural hyperedge aggregation and permutation-invariant set functions (Feng et al., 2019; Zaheer et al., 2017; Lee et al., 2019; Chien et al., 2022). Key-value memory separates an addressing key from its readable value (Miller et al., 2016); MemoRAG uses generated clues to guide a separate evidence search (Qian et al., 2025). Dense X Retrieval further shows that retrieval granularity determines the selectable unit (Chen et al., 2024). HKVM combines these ideas by using bridge-group keys for passage addresses, original passages as values, and a supervised selector under a finite budget.

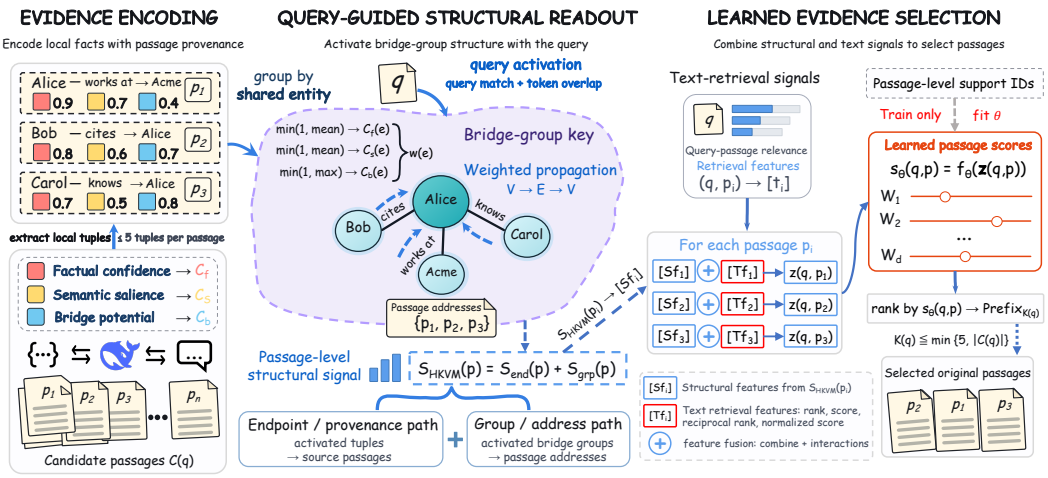

Figure 2: **HKVM method workflow.** Provenance-preserving tuples from candidate passages are grouped around shared entities into bridge-group keys. Query-conditioned weighted propagation returns to the passage addresses along an endpoint path and a group path, whose sum is the passage-level structural signal. A scorer trained with passage-level support IDs combines that signal with text-retrieval features. At inference, the scorer ranks candidates and returns an available prefix of at most five original passages; keys organize access while passage text remains the selected value.

3 METHOD

3.1 RELATIONAL EVIDENCE AS AN ADDRESS–VALUE INTERFACE

Multi-hop support selection is an organization problem before it is a ranking problem. A query may depend on facts distributed across passages, while the final context can contain only a small prefix of the candidate pool. HKVM addresses this tension by separating relational organization from readable content: a key organizes the addresses of passages that participate in a shared structure, the original passage remains the value, and a learned selector decides which values enter the context. Figure 2 summarizes this interface.

For a query q, let $C(q)$ be its candidate passages and let $G(q)$ be the annotated support set. HKVM maps each candidate passage to a score and returns a passage set $\widehat{S}_\theta(q) \subseteq C(q)$ under the available-prefix budget. The tuple cache and relational keys are built from the query and candidate evidence; support IDs enter at the supervision and evaluation stages. The experimental setup instantiates this interface with the fixed candidate pools, support metrics, and comparison axes used in the study.

3.2 PROVENANCE-PRESERVING BRIDGE-GROUP CONSTRUCTION

HKVM begins with local evidence and builds cross-passage organization from the resulting tuple cache. For each candidate passage, the extractor records at most five tuples

$$\tau = (h_\tau, \ell_\tau, t_\tau, p_\tau, c_f(\tau), c_s(\tau), c_b(\tau)), \tag{1}$$

where h_τ and t_τ are entity endpoints, ℓ_τ is a relation phrase, p_τ is the originating passage address, and the three c fields record factual confidence, semantic salience, and bridge potential. Let R denote the resulting tuple cache. The passage address travels with the fact from extraction through structural scoring, so the structure can change how evidence is addressed without changing the evidence text.

The shared entity is the join key that turns local facts into a cross-passage address. For an entity v, HKVM collects its incident tuples

$$G_v = \{\tau \in R : v = h_\tau \text{ or } v = t_\tau\}. \tag{2}$$

When $|G_v| \geq 2$ and the incident tuples contain at least three distinct endpoints, HKVM forms the bridge-group key (a hyperedge connecting passages through a shared entity):

$$e_v = (V_v, P_v), \qquad V_v = \bigcup_{\tau \in G_v} \{h_\tau, t_\tau\}, \qquad P_v = \{p_\tau : \tau \in G_v\}. \tag{3}$$

For $e = e_v$, write $\Gamma_e = G_v$. The key carries a compact summary of the tuples that formed it: factual confidence and semantic salience are averaged, while bridge potential records the strongest bridge-oriented tuple,

$$\begin{aligned} c_f(e) &= \min\left\{1, \frac{1}{|\Gamma_e|} \sum_{\tau \in \Gamma_e} c_f(\tau)\right\}, \\ c_s(e) &= \min\left\{1, \frac{1}{|\Gamma_e|} \sum_{\tau \in \Gamma_e} c_s(\tau)\right\}, \\ c_b(e) &= \min\left\{1, \max_{\tau \in \Gamma_e} c_b(\tau)\right\}. \end{aligned} \tag{4}$$

The resulting key stores the shared-entity organization and the passage addresses participating in it. The passages in P_v remain the values available to the selector or downstream reader. Here, "bridge-group" names the address structure induced by shared entities; support selection is learned after this structure has been built.

3.3 Structural readout to passage addresses

The bridge groups turn local relations into a query-addressable structure. Let $\mathcal{E}$ be the constructed groups and let V_e denote the entity set of group e. Their recorded fields determine the structural weight. We use nonnegative coefficients $(\lambda_f, \lambda_s, \lambda_b)$ for factual confidence, semantic salience, and bridge potential, respectively, and normalize their weighted readout so that the relative scale of the coefficients is immaterial:

$$a(e) = \frac{\lambda_f c_f(e) + \lambda_s c_s(e) + \lambda_b c_b(e)}{\lambda_f + \lambda_s + \lambda_b}, \qquad w(e) = 1 + (\beta - 1)a(e). \tag{5}$$

Because the field scores lie in $[0, 1]$, nonnegative coefficients with positive sum yield $a(e) \in [0, 1]$; for $\beta \geq 1$, the resulting multiplier lies in $[1, \beta]$. Query entity matches and token overlap define the initial entity activation x_v^0. For each vertex incident to a constructed group, let

$$d(v) = \big|\{e \in \mathcal{E} : v \in V_e\}\big|. \tag{6}$$

The query activation then propagates through the bridge groups with restart,

$$x_v^{t+1} = \rho x_v^0 + (1 - \rho) \sum_{e \in \mathcal{E} : v \in V_e} \frac{w(e)}{d(v)|V_e|} \sum_{u \in V_e} x_u^t, \tag{7}$$

where $|V_e|$ is the number of entity vertices in group e, $\rho \in [0, 1]$ controls the restart contribution, and T is a nonnegative integer giving the number of propagation steps. We use one fixed operating point for the HKVM readout across datasets; its values are listed in Appendix F.2, with the operating-point rationale and earlier answer-aware screens summarized in Appendix F.3. After propagation, HKVM returns to the passage addresses through two additive paths. For a passage p, the endpoint contribution and group contribution are

$$\begin{aligned} s_{\mathrm{end}}(p) &= \sum_{\tau \in R :\, p_\tau = p} c_f(\tau)\big(x_{h_\tau}^{(T)} + x_{t_\tau}^{(T)}\big), \\ s_{\mathrm{grp}}(p) &= \sum_{e \in \mathcal{E} :\, p \in P_e} \frac{w(e)}{|V_e|} \sum_{v \in V_e} x_v^{(T)}, \\ s_{\mathrm{HKVM}}(p) &= s_{\mathrm{end}}(p) + s_{\mathrm{grp}}(p). \end{aligned} \tag{8}$$

The endpoint term carries tuple-level evidence back to its source passage. The group term carries the activated key back to every address in P_e. Their sum is the structural signal that the selector sees; the passage value itself remains unchanged.

3.4 LEARNING TO SELECT UNDER A FINITE BUDGET

The bridge group supplies structure; the final decision remains passage-level. For each candidate passage p, the selector forms $\mathbf{z}(q, p)$ from the projected HKVM signal and the available text features: rank, score, reciprocal rank, normalized score, and their interactions. The HKVM variant adds the structural features derived from $s_{\mathrm{HKVM}}(p)$ to this representation. A learned scorer $s_\theta(q, p) = f_\theta(\mathbf{z}(q, p))$ is trained with passage-level support IDs. Structural training predictions are generated out of fold, and fixed predictions are used when scoring development cases.

At inference, the scorer orders the candidate passages and returns the available prefix

$$\widehat{S}_\theta(q) = \mathrm{Prefix}_{K(q)}\left(\mathrm{rank}_{p \in C(q)}\, s_\theta(q, p)\right), \qquad K(q) \le \min\{5, |C(q)|\}. \tag{9}$$

The resulting interface has a clear division of labor: keys organize relational access, values preserve readable evidence, and supervision learns which values deserve the limited context budget. The same ranked prefix can improve average support overlap without improving complete-support retention, so the experimental setup evaluates both views of the selected evidence.

4 EXPERIMENTAL SETUP

4.1 TASK AND EVIDENCE SUBSTRATE

We evaluate the interface defined in Section 3 as a passage-level support-selection task. For each query q, a method receives a candidate pool $C(q)$ and returns a selected prefix $\widehat{S}(q) \subseteq C(q)$. The annotated support set $G(q)$ supplies the target for supervision and evaluation. The measured object is the compact passage set retained within the available context budget, evaluated by how well it covers the supporting evidence required by the query.

Every comparison holds the candidate passages, tuple cache, and passage provenance fixed while changing the structural organization or passage-level controller. This shared evidence substrate isolates how relational structure and learned selection use the same available evidence. The selector returns an available prefix of at most five passages; when a candidate pool contains fewer than five passages, the full available set is retained. For the HKVM readout, we use one shared operating point across datasets: $(\lambda_f, \lambda_s, \lambda_b) = (0.20, 0.40, 0.40)$, $\beta = 2.0$, $\rho = 0.35$, and $T = 1$. Appendix F.3 explains this choice and summarizes the earlier sensitivity checks that informed it.

4.2 DATASETS, METRICS, AND AGGREGATION

We use the development splits of three multi-hop benchmarks that provide passage-level supporting evidence: 2WikiMultiHopQA (12,576 queries) (Ho et al., 2020), the answerable subset of MuSiQue (2,417 queries) (Trivedi et al., 2022), and the distractor setting of HotpotQA (7,405 queries) (Yang et al., 2018). Together they present the same support-selection interface across different multi-hop constructions and candidate difficulties while preserving a common passage-level evaluation target.

For a selected set S and gold support set G, we report support-selection F1 (support F1 below) and support recall,

$$F_{\mathrm{sup}}(S, G) = \frac{2|S \cap G|}{|S| + |G|}, \qquad R_{\mathrm{sup}}(S, G) = \frac{|S \cap G|}{|G|}, \tag{10}$$

and all-support retention,

$$A_{\mathrm{sup}}(S, G) = \mathbb{1}[G \subseteq S]. \tag{11}$$

The first two metrics measure overlap and coverage within the selected set. The third records whether all annotated support passages survive together. We also compute candidate recall and candidate all-support by replacing S with $C(q)$. These candidate-level quantities describe the visibility ceiling of the evidence substrate; the selected-set quantities describe the decisions made within that ceiling.

Each case is evaluated with seeds 13, 29, and 43. We first average each metric across seeds for a given query, then take a query-equal mean within each dataset. This aggregation gives every query equal weight while treating repeated seeds as evaluations of the same case rather than as independent queries.

4.3 COMPARATIVE DESIGN

Our comparative design separates two changes in the evidence-selection pipeline. The *same-source* comparison contrasts a learned controller with the raw ranking produced by the same retrieval source, measuring how passage-level supervision changes the source-specific starting point under a fixed candidate pool and budget. The *cross-source* comparison evaluates the native relational controller alongside learned controllers built from text and alternative structural sources, asking how different evidence organizations guide passage selection after the same learning opportunity.

The learned sources comprise lexical BM25, dense Contriever, late-interaction ColBERTv2, a weighted pairwise PPR graph, a static hypergraph, and a shared-local co-occurrence hypergraph. We refer to the two relational variants as HippoRAG-PPR and Hyper-RAG-HG. HippoRAG-PPR instantiates the graph-to-passage organization associated with HippoRAG (Gutiérrez et al., 2024), while Hyper-RAG-HG instantiates the shared-local relational organization exemplified by Hyper-RAG (Feng et al., 2026); both operate on the common tuple cache and candidate substrate. Here, "same learning opportunity" means the same candidate substrate, support-ID supervision, selection budget, seed aggregation, and comparison protocol; it does not assert equal model capacity or equal upstream supervision. The main results report the complete same-pool contrasts across the three selected-set metrics. The separate set-level diagnostic uses a different candidate substrate and learning target and is interpreted alongside the main comparison as a mechanism analysis.

5 EXPERIMENTAL RESULTS AND ANALYSIS

The evaluation separates two changes that are often conflated in structured retrieval. First, passage-level supervision changes the ranking produced by any retrieval source. Second, after controlling for this learning opportunity, relational organization may or may not improve the selected evidence relative to another learned source. We report these comparisons on the same candidate substrate and then examine how support objectives, candidate visibility, group composition, and the form of the structural readout shape the observed pattern.

5.1 PASSAGE-LEVEL LEARNING SHIFTS EVERY SOURCE'S BASELINE

Before comparing evidence organizations, we first isolate the change introduced by passage-level supervision. The same-source contrast is positive for all 18 source–dataset cells. Support-selection F1 increases by 2.06–$14.48\,pp$, with the smallest gain for Contriever on MuSiQue and the largest for Contriever on 2Wiki; all-support retention also increases in every cell, with gains of 3.30–$42.13\,pp$. The pattern spans lexical, dense, late-interaction, pairwise-relational, and grouped-relational sources. A raw ranking is therefore not an adequate baseline for comparing learned controllers: the cross-source question must be evaluated after the learning opportunity has been matched.

5.2 AFTER LEARNING, RELATIONAL DIFFERENCES REMAIN CONDITIONAL

With this calibration in place, the relevant comparison is between learned controllers. Table 1 reports the complete absolute comparison of the native relational controller and six learned alternatives on the same candidate substrate, with support F1, support recall, and all-support retention shown separately for each dataset. The leading controller changes across the three datasets even though every method receives the same candidate evidence and budget.

The absolute values expose a conditional pattern that a single reference would hide. On 2Wiki, Static-HG is marginally highest on support F1 and recall, whereas HKVM has the highest all-support retention. On MuSiQue, Contriever has the highest support F1, while HKVM has the highest all-support retention and is tied with Contriever on recall at the displayed precision. On HotpotQA, BM25 is highest on all three metrics. The native relational controller is therefore competitive across the learned comparison, but its position changes with both the comparator and the support objective rather than defining a universal ranking.

Figure 3 shows both comparisons together: the learning shift above, the matched relational contrasts below on a shared zero-centered scale.

Table 1: **Absolute support-selection results under fixed candidate pools.** Entries are percentages. Each dataset reports support F1, support recall, and all-support retention (All). Bold marks the largest value for each dataset–metric pair, with ties after rounding jointly bolded. All methods use the same candidate substrate and up-to-five available-prefix budget.

	2Wiki			MuSiQue			HotpotQA		
Method	F1	Rec.	All	F1	Rec.	All	F1	Rec.	All
Text retrieval									
BM25	60.92	95.33	87.93	42.65	64.26	30.95	**53.40**	**93.19**	**86.58**
Contriever	61.24	95.82	89.08	**43.86**	**65.92**	33.49	51.93	90.61	81.81
ColBERTv2	60.58	94.70	86.77	42.16	63.66	31.14	51.07	89.10	78.81
Relational organizations									
HippoRAG-PPR	61.92	96.62	91.87	41.99	63.46	30.20	51.78	90.35	81.10
Hyper-RAG-HG	61.82	96.51	91.42	41.89	63.27	30.01	51.41	89.71	79.92
Static-HG	**62.25**	**97.09**	93.13	42.00	63.52	30.58	51.58	89.99	80.46
HKVM (native)	62.20	97.01	**93.31**	43.68	**65.92**	**34.71**	51.43	89.73	79.96

5.3 Support objectives expose different selection targets

The selected-set metrics do not measure the same behavior. On MuSiQue, native minus Contriever is $-0.18\,pp$ in support F1 and $+0.001\,pp$ in recall, yet $+1.23\,pp$ in all-support retention. On 2Wiki, native minus Static-HG is $-0.05\,pp$ in F1 and $-0.08\,pp$ in recall, while all-support retention is $+0.18\,pp$. By contrast, the HotpotQA–BM25 comparison is negative on all three metrics. Average overlap and complete-support retention can therefore rank the same selected sets differently; both are needed to characterize what the selector preserves.

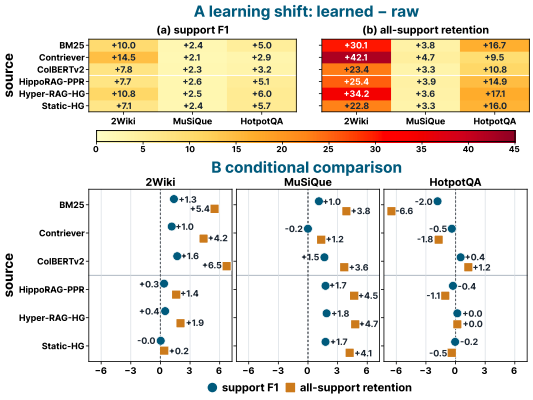

Figure 3: **Learning shifts raw baselines; relational differences remain conditional.** (A) Raw-to-learned changes for support F1 and all-support retention. (B) Native relational controller minus each learned alternative; circles denote F1; squares, all-support retention. All contrasts use the same candidate pool, query-equal aggregation, and up-to-five available-prefix budget. Support recall is reported in Table 1 and Appendix B.

5.4 Candidate visibility and complete support

The candidate pool places a ceiling on every selected-set result. Candidate recall is 100.00%, 76.54%, and 100.00% on 2Wiki, MuSiQue, and HotpotQA, while candidate all-support is 100.00%, 49.32%, and 100.00%, respectively. The native selector retains all annotated support passages for 93.31%, 34.71%, and 79.96% of queries. The gap between candidate and selected all-support is $6.69\,pp$, $14.61\,pp$, and $20.04\,pp$ on the three datasets, and measures loss of complete support within the visible pool. Missing evidence accounts for the separate shortfall of candidate all-support from 100%; the candidate-to-selected gap concerns queries whose full support is visible but is not retained under the budgeted decision. Candidate completeness is a property of the evidence substrate, whereas selected all-support measures the controller's decision within that substrate.

5.5 Group signal and distractors

The group construction contains useful support-related signal, but it does not produce clean support sets by itself. On 256 deterministically selected training cases per dataset, the best gold-defined group completion has mean support F1 of 62.86%, 46.87%, and 52.52% on 2Wiki, MuSiQue, and HotpotQA. Among multi-passage groups that hit at least one support passage, the fractions of groups containing at least one non-support passage are 40.79%, 64.98%, and 80.47%. These values come from a group-composition diagnostic that uses gold support to ask how much signal is present in

Table 2: **Diagnostic views of candidate visibility, group composition, and mechanism boundary.** *Visibility*: candidate and selected all-support retention on the fixed candidate pool (all development queries). *Group composition*: maximum support F1 and distractor fraction from gold-defined group completion (256 deterministically selected training cases per dataset). *Set-level comparison*: group-completion-minus-pairwise support F1 change with paired-bootstrap confidence intervals (BM25 top-10, 512 fitting / 128 evaluation cases per dataset). The propagation check is described in the appendix.

	Visibility		*Group composition*		*Set-level comparison*	
Dataset	Cand.	Sel.	Max F1	Dist.	Group−pairwise	95% CI
2Wiki	100%	93.3%	62.9%	40.8%	$-0.46\,pp$	$[-1.50, +0.34]$ pp
MuSiQue	49.3%	34.7%	46.9%	65.0%	$-0.36\,pp$	$[-1.20, +0.42]$ pp
HotpotQA	100%	80.0%	52.5%	80.5%	$+0.23\,pp$	$[-1.08, +1.57]$ pp

a group; they motivate a learned selector that must preserve useful addresses while rejecting co-occurring material.

5.6 Representation, propagation, and learning-target boundary

The propagation check separates the structural representation from the particular propagation formula used to read it. Across 384 matched comparisons, the implemented diffusion at $T \in \{1, 2\}$ and its passage projection agree with the corresponding normalized-incidence form to a maximum relative residual of 7.38×10^{-16} and a maximum absolute residual of 1.07×10^{-14}. The structural readout is therefore a linear realization of the address signal defined in Method; the evidence does not require an irreducible higher-order interpretation.

An independent set-level comparison asks whether explicit group-completion features improve direct learning of a passage set on BM25 top-10 candidates. With 512 fitting and 128 evaluation cases per dataset, group-completion-minus-pairwise support F1 changes are -0.46, -0.36, and $+0.23\,pp$ on 2Wiki, MuSiQue, and HotpotQA, with paired-bootstrap 95% confidence intervals of $[-1.50, +0.34]$ pp, $[-1.20, +0.42]$ pp, and $[-1.08, +1.57]$ pp. All intervals cross zero, and the Holm-adjusted p-value is .9941. Taken together with the group composition result, this locates the contribution at the representation–selection interface: relational groups can carry useful evidence, while the added value of a particular explicit set head is not established by this parameterization.

Figure 4 and Table 2 collect these three views—candidate visibility, group composition, and the learning target—each under its own candidate and learning protocol.

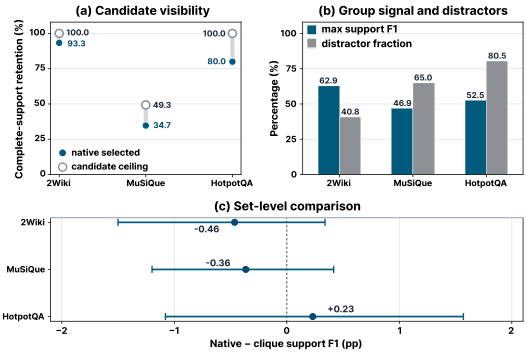

Figure 4: **Conditional diagnostics.** (a) Candidate versus native selected all-support, separating pool visibility from selection. (b) Gold-defined group-composition maximum F1 and distractor fraction on 256 training cases per dataset. (c) Group-completion-minus-pairwise F1 with paired-bootstrap intervals on BM25 top-10 candidates. The panels use different protocols and are not a single performance curve.

6 Discussion

HKVM frames relational retrieval as an evidence-organization layer that a learned selector can query. The experimental results show why such an interface must be read jointly with its candidate substrate, learned comparator, and support objective, rather than as a single verdict on relational organization.

The findings separate learning effects from organizational effects. Across six retrieval sources, moving from raw ranking to a learned controller consistently improves support F1 on all three datasets, with gains ranging from roughly two to fourteen percentage points. This uniform direction shows that supervised passage-level selection changes the baseline for

comparison. Against that learned baseline, the native relational controller shows clear utility on some contrasts—most consistently when the learned alternative is built from the same dense retriever that also supplies the candidate pool—and near-zero or negative utility on others, including some BM25- and Contriever-based comparisons on HotpotQA. The conditional pattern reflects real differences in candidate visibility, in how structure interacts with the learned representation, and in what each support objective rewards.

Candidate visibility sets an upper bound on selected support: when gold-support passages are absent from the candidate pool, no selector can retrieve them. MuSiQue shows systematically lower candidate all-support retention than 2Wiki or HotpotQA, and the native controller's selected all-support remains below that ceiling on all three datasets. Even where relational organization identifies the relevant passage groups, the selector may still drop some of them if the support objective or the learned representation does not favor complete coverage. Separating candidate completeness from selected completeness therefore locates the bottleneck: missing evidence and dropped evidence are distinct failure modes, and improving one does not fix the other.

The group-composition and propagation diagnostics further unpack what relational organization contributes. The group-composition diagnostic shows that constructed bridge-group hyperedges can contain support-related signal while also introducing distractors—on 2Wiki the gold-defined group completion reaches approximately 0.63 support F1 with a distractor fraction near 0.41, meaning useful structure and noise coexist in the same groups. The propagation check confirms that the current linear propagation and projection can be matched by the matched normalized-incidence update and passage projection, establishing the representation–operator boundary within this parameterization. The exploratory set-level comparison, which directly learns passage sets with explicit interactions, has paired-bootstrap intervals that cross zero on all three datasets, so the specific parameterization tested does not establish a robust set-level advantage. These diagnostics do not refute the value of relational organization in general—they describe the boundaries of what the current design and learning setup achieve, and they provide reusable concepts—candidate visibility, group signal and distractors, representation versus operator effects—for future passage-selection work to build on.

This conditionality is the study's empirical lesson rather than a shortfall to be defended: the question is not whether hypergraph retrieval universally outperforms all learned alternatives, but when and how relational organization changes passage selection once learning has already lifted the raw baseline. Making the key–value separation explicit, and isolating learning effects from organizational effects, gives that question a form others can test, refine, or replace.

7 Conclusion

HKVM makes the relation-address and passage-content distinction explicit at the point where a learned selector must allocate a finite evidence budget: bridge-group hyperedges organize passage provenance through shared entities, the original passages remain the values selected downstream, and supervision decides which of them to retain. The resulting empirical picture is conditional: learning consistently changes the raw-source baseline across six retrieval sources and three datasets, while the utility of relational organization relative to other learned alternatives depends on candidate visibility, the learned comparator, and the support target being optimized.

The study contributes an explicit address–value interface for multi-hop evidence selection, a diagnostic vocabulary that separates candidate visibility from selected support and that distinguishes group signal from distractors, and a set of matched comparisons that isolate learning effects from organizational effects. Structured organization emerges as one instrument for passage selection among several, not as a replacement for learned retrieval or a guarantee of complete-support retention.

The diagnostic framing introduced here—candidate completeness as a visibility ceiling, group composition as a construction diagnostic, propagation matching as a representation boundary, and independent set-level parameterizations as learning-target comparisons—applies to future multi-hop evidence work regardless of the specific hypergraph design. The current study tests these ideas on development splits with fixed candidate pools and offline support IDs; extending them to answer generation, to full-corpus retrieval, or to online candidate construction would require new evidence and opens, rather than closes, the question of how structure should interact with learning in multi-hop retrieval.

REFERENCES

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense X retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 15159–15177, 2024. doi: 10.18653/v1/2024.emnlp-main.845.

Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. You are allset: A multiset function framework for hypergraph neural networks. In *ICLR*, 2022.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *AAAI*, 2019.

Yifan Feng, Hao Hu, Shihui Ying, Xingliang Hou, Shiquan Liu, Mingyuan Yang, Junchang Li, Shaoyi Du, Nanning Zheng, Han Hu, and Yue Gao. Hyper-RAG: Combating LLM hallucinations using hypergraph-driven retrieval-augmented generation. *Nature Communications*, 2026. doi: 10.1038/s41467-026-71411-1.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 59532–59569, 2024.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 2025.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-Retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Xanh Ho et al. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *COLING*, 2020.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6944–6961, 2024.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji-Rong Wen. StructGPT: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9237–9251, 2023.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. Baleen: Robust multi-hop reasoning at scale via condensed retrieval. In *Advances in Neural Information Processing Systems*, volume 34, pp. 27670–27682, 2021.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020.

Haoran Luo, Guanting Chen, Yandan Zheng, Xiaobao Wu, Yikai Guo, Qika Lin, Yu Feng, Zemin Kuang, Meina Song, Yifan Zhu, et al. Hypergraphrag: Retrieval-augmented generation via hypergraph-structured knowledge representation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 152206–152234, 2026.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *EMNLP*, 2016.

Hongjin Qian, Zheng Liu, Peitian Zhang, Kelong Mao, Defu Lian, Zhicheng Dou, and Tiejun Huang. Memorag: Boosting long context processing with global memory-enhanced retrieval augmentation. In *Proceedings of the ACM on Web Conference 2025*, pp. 2366–2377, 2025.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3715–3734, 2022.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations*, 2024.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10046, 2023.

Harsh Trivedi et al. Musique: Multihop questions via single-hop question composition. *TACL*, 2022.

Changjian Wang, Weihong Deng, Weili Guan, Quan Lu, and Ning Jiang. Cross-granularity hypergraph retrieval-augmented generation for multi-hop question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33368–33376, 2026. doi: 10.1609/aaai.v40i39.40623.

Zhilin Yang et al. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *EMNLP*, 2018.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *NeurIPS*, 2017.

A Protocol and Reproducibility

We evaluate support identification on the development splits of three datasets: 12,576 cases from 2WikiMultiHopQA, 2,417 answerable cases from MuSiQue, and 7,405 distractor cases from HotpotQA. Each comparison uses three random seeds (13, 29, and 43). For each query, we average metric values across seeds before computing dataset-level aggregates. Within each comparison, both conditions use the same candidate pool; the selector may return up to five passages. We retain queries with fewer than five candidates rather than discarding them.

For selected set S and gold support G, we report $F_1(S, G)$, $|S \cap G|/|G|$, and $\mathbb{1}[G \subseteq S]$. Candidate recall and candidate all-support replace S by the candidate pool C. Absolute table entries are percentages and contrast entries are percentage points. The matrix consists of descriptive means; no confidence intervals or significance tests are reported.

For compactness, the complete comparison uses two families of codes. A1–A6 are same-source learning comparisons, in the fixed source order ColBERTv2, Hyper-RAG-HG, Static-HG, HippoRAG-PPR, BM25, and Contriever; each value is the source-calibrated condition minus that source's raw ranking. B1–B6 use the same source order; each value is the native HKVM controller minus the source-specific learned comparator. Thus, the letter identifies the comparison family and the number identifies the retrieval source, rather than a model version or an additional experimental condition. All 36 comparisons satisfy the same-pool eligibility rule. The Hyper-RAG-HG label denotes the shared-local relational source used here, rather than an external end-to-end reproduction.

All ICLR conditions reuse one fixed passage-level extraction cache. The cache was produced once with DeepSeek-V4-Flash at temperature 0 using a fixed provenance-preserving six-field tuple schema; it is shared across datasets and experimental conditions rather than regenerated for a particular comparison. The cache contains 12,576 2Wiki development cases, 2,417 answerable MuSiQue cases, and 7,405 HotpotQA distractor cases, together with the canonical 10k training subsets used for fitting. Each passage contributes at most five tuples, and each tuple retains its passage provenance and factual-confidence, semantic-salience, and bridge-potential fields. The extraction cache is therefore part of the fixed evidence substrate, not an additional experimental degree of freedom.

A.1 Candidate and selection checks

The support-ID reconstruction checks passage identities, candidate views, and final rank orders from the saved outputs. A valid selected prefix contains unique passage IDs, all within its candidate pool, and consists of the first at most five IDs in the final ranking. The compared conditions must also share the same candidate view. The archive covers 44 source–condition configurations across three datasets and three seeds; the paper uses 19 matched conditions and 36 comparison–dataset cells. The remaining configurations are retained as descriptive records, not as additional matched evidence. All 36 paper-facing cells pass the candidate-view checks.

The 108 comparison–dataset–seed summaries are reduced to 36 dataset-level comparisons by averaging seeds within each query before taking a query-equal mean. Shared passages create cross-query dependence, so the main comparison reports descriptive means on these development queries. The separate bootstrap analysis in Appendix C belongs to the independently defined set-level comparison.

B Complete Matched Comparisons

Table 3 contains all 36 matched comparisons and all three support metrics. The comparison codes and their directions are defined above; positive values favor the first condition in each family. Values are percentage points.

Table 3: Complete matched comparisons. Each code identifies a retrieval source and comparison family as defined in the text. Entries are differences in support F_1, recall, and all-support retention (percentage points); positive values favor the source-calibrated condition in A1–A6 and the native HKVM controller in B1–B6.

Code	Source / dataset	ΔF_1	Δ Recall	Δ All-support
Same-source learning: source-calibrated condition minus raw ranking				
A1	ColBERTv2 / 2Wiki	+7.7776	+10.8348	+23.3593
	ColBERTv2 / MuSiQue	+2.3416	+3.2881	+3.3099
	ColBERTv2 / HotpotQA	+3.2038	+5.6066	+10.8080
A2	Hyper-RAG-HG / 2Wiki	+10.7966	+15.8934	+34.1868
	Hyper-RAG-HG / MuSiQue	+2.4946	+3.7133	+3.6271
	Hyper-RAG-HG / HotpotQA	+5.9599	+10.4299	+17.1371
A3	Static-HG / 2Wiki	+7.1165	+10.5028	+22.7947
	Static-HG / MuSiQue	+2.4137	+3.6627	+3.2961
	Static-HG / HotpotQA	+5.6937	+9.9640	+16.0477
A4	HippoRAG-PPR / 2Wiki	+7.7300	+11.4391	+25.4188
	HippoRAG-PPR / MuSiQue	+2.6042	+3.9282	+3.9443
	HippoRAG-PPR / HotpotQA	+5.1458	+9.0052	+14.9178
A5	BM25 / 2Wiki	+9.9615	+14.3647	+30.0679
	BM25 / MuSiQue	+2.4154	+3.4478	+3.8340
	BM25 / HotpotQA	+5.0133	+8.7734	+16.6824
A6	Contriever / 2Wiki	+14.4766	+21.3846	+42.1305
	Contriever / MuSiQue	+2.0636	+3.1157	+4.6752
	Contriever / HotpotQA	+2.9234	+5.1159	+9.5251
Cross-source comparison: native HKVM controller minus learned alternative				
B1	ColBERTv2 / 2Wiki	+1.6142	+2.3007	+6.5416
	ColBERTv2 / MuSiQue	+1.5233	+2.2583	+3.5719
	ColBERTv2 / HotpotQA	+0.3588	+0.6280	+1.1569
B2	Hyper-RAG-HG / 2Wiki	+0.3800	+0.4957	+1.8925
	Hyper-RAG-HG / MuSiQue	+1.7915	+2.6548	+4.7028
	Hyper-RAG-HG / HotpotQA	+0.0129	+0.0225	+0.0405
B3	Static-HG / 2Wiki	-0.0497	-0.0815	+0.1802
	Static-HG / MuSiQue	+1.6744	+2.4020	+4.1374
	Static-HG / HotpotQA	-0.1518	-0.2656	-0.4997
B4	HippoRAG-PPR / 2Wiki	+0.2761	+0.3896	+1.4392
	HippoRAG-PPR / MuSiQue	+1.6900	+2.4571	+4.5097
	HippoRAG-PPR / HotpotQA	-0.3537	-0.6190	-1.1389
B5	BM25 / 2Wiki	+1.2799	+1.6718	+5.3753
	BM25 / MuSiQue	+1.0307	+1.6630	+3.7650
	BM25 / HotpotQA	-1.9768	-3.4594	-6.6126
B6	Contriever / 2Wiki	+0.9608	+1.1855	+4.2329
	Contriever / MuSiQue	-0.1802	+0.0011	+1.2274
	Contriever / HotpotQA	-0.5067	-0.8868	-1.8456

Table 4 reports the absolute values for every condition used in the matched comparisons. Each cell is support-selection F1/recall/all-support retention in percent after averaging seeds within each query.

Table 4: Absolute results for every compared condition (support-selection F1/recall/all-support retention, %). Raw denotes the original source ranking, calibrated denotes the source-specific learned condition used in A1–A6, learned comparator denotes the source-specific learned condition used in B1–B6, and native denotes the HKVM controller.

Source / condition	2Wiki	MuSiQue	HotpotQA
ColBERTv2 raw	52.9944/ 84.1524/ 64.1142	39.8034/ 60.3641/ 27.6789	47.9294/ 83.6057/ 68.1972
ColBERTv2 calibrated	60.7720/ 94.9871/ 87.4735	42.1450/ 63.6521/ 30.9888	51.1331/ 89.2122/ 79.0052

Source / condition	2Wiki	MuSiQue	HotpotQA
ColBERTv2 learned comparator	60.5838/ 94.7049/ 86.7684	42.1558/ 63.6625/ 31.1405	51.0688/ 89.0997/ 78.8071
Hyper-RAG-HG raw	50.9047/ 80.4369/ 56.7748	35.6539/ 53.9098/ 20.0248	43.7392/ 76.2728/ 57.7178
Hyper-RAG-HG calibrated	61.7013/ 96.3303/ 90.9616	38.1484/ 57.6231/ 23.6519	49.6991/ 86.7027/ 74.8548
Hyper-RAG-HG learned comparator	61.8180/ 96.5099/ 91.4175	41.8877/ 63.2660/ 30.0097	51.4148/ 89.7052/ 79.9235
Static-HG raw	55.0903/ 86.5160/ 70.1892	36.1756/ 54.7855/ 22.2176	44.4337/ 77.4882/ 60.1891
Static-HG calibrated	62.2068/ 97.0188/ 92.9840	38.5893/ 58.4483/ 25.5137	50.1274/ 87.4522/ 76.2368
Static-HG learned comparator	62.2477/ 97.0870/ 93.1298	42.0048/ 63.5188/ 30.5751	51.5794/ 89.9932/ 80.4637
HippoRAG-PPR raw	54.1679/ 85.1403/ 66.3725	35.8057/ 54.2029/ 20.6868	45.0356/ 78.5415/ 61.4855
HippoRAG-PPR calibrated	61.8979/ 96.5795/ 91.7912	38.4100/ 58.1311/ 24.6311	50.1814/ 87.5467/ 76.4033
HippoRAG-PPR learned comparator	61.9219/ 96.6159/ 91.8708	41.9891/ 63.4637/ 30.2027	51.7814/ 90.3466/ 81.1029
HKVM native controller	62.1980/ 97.0055/ 93.3100	43.6791/ 65.9208/ 34.7125	51.4277/ 89.7277/ 79.9640
BM25 raw	50.6111/ 80.3992/ 56.6555	38.0806/ 57.4886/ 23.1278	47.7442/ 83.2816/ 67.7380
BM25 calibrated	60.5727/ 94.7638/ 86.7234	40.4960/ 60.9364/ 26.9618	52.7575/ 92.0549/ 84.4204
BM25 learned comparator	60.9181/ 95.3337/ 87.9347	42.6484/ 64.2578/ 30.9475	53.4045/ 93.1870/ 86.5766
Contriever raw	46.6075/ 74.1969/ 46.3979	41.4733/ 62.2880/ 28.1341	48.6625/ 84.8886/ 71.3302
Contriever calibrated	61.0841/ 95.5815/ 88.5284	43.5369/ 65.4036/ 32.8093	51.5859/ 90.0045/ 80.8553
Contriever learned comparator	61.2372/ 95.8201/ 89.0771	43.8594/ 65.9196/ 33.4850	51.9344/ 90.6144/ 81.8096

The following table lists the three seed-level summaries behind each matched comparison. It is included to make the aggregation transparent; it is not used to manufacture uncertainty estimates because shared passages induce substantial cross-query dependence in two of the three datasets. In A1–A6, the primary condition is the source-calibrated condition and the comparator is the raw source. In B1–B6, the primary condition is native HKVM and the comparator is the corresponding learned comparator; every delta is primary minus comparator.

Table 5: Seed-level summaries for the matched comparisons. The code uses the mapping in Table 3; values are fractions, and the final three columns are primary-minus-comparator deltas.

Code	Dataset	Seed	n	Primary F1	Comparator F1	ΔR	Δ all
A1	2Wiki	13	12576	0.607448	0.529944	0.107626	0.232109
A1	2Wiki	29	12576	0.605855	0.529944	0.105876	0.226861
A1	2Wiki	43	12576	0.609857	0.529944	0.111542	0.241810
A1	MuSiQue	13	2417	0.422201	0.398034	0.033616	0.031858
A1	MuSiQue	29	2417	0.422328	0.398034	0.034340	0.036409
A1	MuSiQue	43	2417	0.419821	0.398034	0.030685	0.031030
A1	HotpotQA	13	7405	0.512669	0.479294	0.058406	0.112762
A1	HotpotQA	29	7405	0.510856	0.479294	0.055233	0.106550
A1	HotpotQA	43	7405	0.510470	0.479294	0.054558	0.104929
A2	2Wiki	13	12576	0.616453	0.509047	0.158119	0.339695
A2	2Wiki	29	12576	0.617435	0.509047	0.159550	0.343511
A2	2Wiki	43	12576	0.617152	0.509047	0.159132	0.342398
A2	MuSiQue	13	2417	0.381302	0.356539	0.036857	0.036823
A2	MuSiQue	29	2417	0.382545	0.356539	0.038443	0.038477
A2	MuSiQue	43	2417	0.380606	0.356539	0.036098	0.033513
A2	HotpotQA	13	7405	0.497158	0.437392	0.104591	0.171911
A2	HotpotQA	29	7405	0.496271	0.437392	0.103038	0.168805
A2	HotpotQA	43	7405	0.497544	0.437392	0.105267	0.173396
A3	2Wiki	13	12576	0.622390	0.550903	0.105499	0.229246
A3	2Wiki	29	12576	0.621935	0.550903	0.104902	0.227258
A3	2Wiki	43	12576	0.621880	0.550903	0.104684	0.227338
A3	MuSiQue	13	2417	0.384343	0.361756	0.033823	0.030616
A3	MuSiQue	29	2417	0.388488	0.361756	0.040374	0.039719
A3	MuSiQue	43	2417	0.384848	0.361756	0.035685	0.028548
A3	HotpotQA	13	7405	0.501441	0.444337	0.099932	0.161107
A3	HotpotQA	29	7405	0.501017	0.444337	0.099190	0.160162
A3	HotpotQA	43	7405	0.501364	0.444337	0.099797	0.160162
A4	2Wiki	13	12576	0.618851	0.541679	0.114365	0.253737
A4	2Wiki	29	12576	0.618762	0.541679	0.113967	0.253260
A4	2Wiki	43	12576	0.619325	0.541679	0.114842	0.255566
A4	MuSiQue	13	2417	0.383811	0.358057	0.039064	0.039719
A4	MuSiQue	29	2417	0.384684	0.358057	0.039960	0.040960
A4	MuSiQue	43	2417	0.383804	0.358057	0.038822	0.037650
A4	HotpotQA	13	7405	0.501981	0.450356	0.090344	0.150844
A4	HotpotQA	29	7405	0.501094	0.450356	0.088791	0.146253
A4	HotpotQA	43	7405	0.502367	0.450356	0.091020	0.150439
A5	2Wiki	13	12576	0.605931	0.506111	0.144004	0.301527
A5	2Wiki	29	12576	0.605005	0.506111	0.142494	0.297789
A5	2Wiki	43	12576	0.606244	0.506111	0.144442	0.302719
A5	MuSiQue	13	2417	0.406354	0.380806	0.036305	0.038477
A5	MuSiQue	29	2417	0.406589	0.380806	0.037133	0.044270
A5	MuSiQue	43	2417	0.401936	0.380806	0.029996	0.032271
A5	HotpotQA	13	7405	0.527755	0.477442	0.088049	0.167319
A5	HotpotQA	29	7405	0.527138	0.477442	0.086968	0.165159
A5	HotpotQA	43	7405	0.527833	0.477442	0.088184	0.167995
A6	2Wiki	13	12576	0.610927	0.466075	0.214019	0.421835
A6	2Wiki	29	12576	0.610690	0.466075	0.213482	0.420086
A6	2Wiki	43	12576	0.610907	0.466075	0.214039	0.421994
A6	MuSiQue	13	2417	0.435641	0.414733	0.031547	0.049235
A6	MuSiQue	29	2417	0.435242	0.414733	0.030720	0.045925
A6	MuSiQue	43	2417	0.435222	0.414733	0.031203	0.045097
A6	HotpotQA	13	7405	0.515756	0.486625	0.050979	0.094801
A6	HotpotQA	29	7405	0.516450	0.486625	0.052194	0.097502
A6	HotpotQA	43	7405	0.515370	0.486625	0.050304	0.093450
B1	2Wiki	13	12576	0.622617	0.607711	0.021569	0.060989
B1	2Wiki	29	12576	0.621668	0.605810	0.022285	0.064090
B1	2Wiki	43	12576	0.621655	0.603992	0.025167	0.071167
B1	MuSiQue	13	2417	0.439552	0.422357	0.025134	0.035995
B1	MuSiQue	29	2417	0.434845	0.422937	0.018239	0.030203
B1	MuSiQue	43	2417	0.435978	0.419381	0.024376	0.040960

Code	Dataset	Seed	n	Primary F1	Comparator F1	ΔR	Δ all
B1	HotpotQA	13	7405	0.512939	0.511589	0.002363	0.003646
B1	HotpotQA	29	7405	0.514482	0.511357	0.005469	0.009993
B1	HotpotQA	43	7405	0.515409	0.509119	0.011006	0.021067
B2	2Wiki	13	12576	0.622617	0.617745	0.006560	0.022901
B2	2Wiki	29	12576	0.621668	0.618157	0.004632	0.017573
B2	2Wiki	43	12576	0.621655	0.618639	0.003678	0.016301
B2	MuSiQue	13	2417	0.439552	0.420484	0.027996	0.045925
B2	MuSiQue	29	2417	0.434845	0.419991	0.022445	0.040546
B2	MuSiQue	43	2417	0.435978	0.416154	0.029203	0.054613
B2	HotpotQA	13	7405	0.512939	0.513827	-0.001553	-0.003241
B2	HotpotQA	29	7405	0.514482	0.514097	0.000675	0.001621
B2	HotpotQA	43	7405	0.515409	0.514521	0.001553	0.002836
B3	2Wiki	13	12576	0.622617	0.622531	0.000139	0.003658
B3	2Wiki	29	12576	0.621668	0.622137	-0.000755	0.001988
B3	2Wiki	43	12576	0.621655	0.622763	-0.001829	-0.000239
B3	MuSiQue	13	2417	0.439552	0.423040	0.023135	0.036823
B3	MuSiQue	29	2417	0.434845	0.419543	0.022583	0.038891
B3	MuSiQue	43	2417	0.435978	0.417560	0.026341	0.048407
B3	HotpotQA	13	7405	0.512939	0.515177	-0.003916	-0.007562
B3	HotpotQA	29	7405	0.514482	0.515987	-0.002633	-0.004997
B3	HotpotQA	43	7405	0.515409	0.516219	-0.001418	-0.002431
B4	2Wiki	13	12576	0.622617	0.619219	0.004731	0.016619
B4	2Wiki	29	12576	0.621668	0.618396	0.004930	0.016221
B4	2Wiki	43	12576	0.621655	0.620042	0.002028	0.010337
B4	MuSiQue	13	2417	0.439552	0.419280	0.029341	0.049235
B4	MuSiQue	29	2417	0.434845	0.421461	0.020032	0.038891
B4	MuSiQue	43	2417	0.435978	0.418932	0.024341	0.047166
B4	HotpotQA	13	7405	0.512939	0.517492	-0.007968	-0.015530
B4	HotpotQA	29	7405	0.514482	0.516798	-0.004051	-0.007157
B4	HotpotQA	43	7405	0.515409	0.519151	-0.006550	-0.011479
B5	2Wiki	13	12576	0.622617	0.609220	0.017812	0.055344
B5	2Wiki	29	12576	0.621668	0.607908	0.018249	0.057490
B5	2Wiki	43	12576	0.621655	0.610414	0.014094	0.048426
B5	MuSiQue	13	2417	0.439552	0.427488	0.019446	0.039719
B5	MuSiQue	29	2417	0.434845	0.427220	0.012653	0.030203
B5	MuSiQue	43	2417	0.435978	0.424744	0.017791	0.043029
B5	HotpotQA	13	7405	0.512939	0.534199	-0.037205	-0.071573
B5	HotpotQA	29	7405	0.514482	0.533774	-0.033761	-0.064416
B5	HotpotQA	43	7405	0.515409	0.534160	-0.032816	-0.062390
B6	2Wiki	13	12576	0.622617	0.612444	0.012723	0.043814
B6	2Wiki	29	12576	0.621668	0.611629	0.012544	0.044529
B6	2Wiki	43	12576	0.621655	0.613045	0.010297	0.038645
B6	MuSiQue	13	2417	0.439552	0.438020	0.004620	0.016549
B6	MuSiQue	29	2417	0.434845	0.439828	-0.003965	0.005792
B6	MuSiQue	43	2417	0.435978	0.437933	-0.000621	0.014481
B6	HotpotQA	13	7405	0.512939	0.519884	-0.012154	-0.024983
B6	HotpotQA	29	7405	0.514482	0.518842	-0.007630	-0.016070
B6	HotpotQA	43	7405	0.515409	0.519305	-0.006820	-0.014315

C Conditional Diagnostics and Set-Level Comparison

These diagnostics use distinct samples and units. Candidate visibility is measured on the main development pools; group composition is examined on training cases; and the set-level comparison uses a separate internal fitting and evaluation split.

C.1 Candidate visibility

Table 6 summarizes candidate visibility and native selected support on the main development pools. The set-level comparison is intentionally separate and uses BM25 top-10 candidates.

Table 6: Candidate pool statistics from development splits

Statistic	**2Wiki**	**MuSiQue**	**HotpotQA**
Questions (dev split)	12,576	2,417	7,405
Candidate recall (%)	100.00	76.54	100.00
Candidate all-support (%)	100.00	49.32	100.00
Selected all-support (%)	93.31	34.71	79.96
Gap (candidate - selected)	6.69 pp	14.61 pp	20.04 pp

Table 7: Group-composition diagnostic: mean maximum support F1 (%) over groups of two to five passages. Every condition uses the same 256 training queries per dataset; seeds index randomized controls rather than repeated learned fits.

Group addresses	Seed	2Wiki	MuSiQue	HotpotQA
Native groups	—	62.86	46.87	52.52
Global permutation	13	34.77	26.99	33.71
Global permutation	29	35.27	24.17	32.70
Global permutation	43	32.45	24.66	32.37
Degree-preserving swap	13	36.92	27.95	38.22
Degree-preserving swap	29	37.99	28.34	37.46
Degree-preserving swap	43	37.11	27.28	37.14

C.2 Group composition and randomized controls

The group-composition diagnostic uses 256 training cases per dataset selected by a deterministic order and the same extracted tuple cache. For each query, it computes the maximum support F1 over constructed groups addressing two to five passages, assigning zero if no eligible group exists. The mean includes all 256 cases. Gold support identifies the best group only for this diagnostic; it is not an inference-time selection rule.

The distractor measure has a different unit and eligibility rule. Let $\mathcal{H}_q$ contain all groups with at least two passages and at least one annotated support passage. The reported fraction is

$$D_{\text{group}} = \frac{\sum_q \sum_{e \in \mathcal{H}_q} \mathbb{1}[P_e \setminus G(q) \neq \emptyset]}{\sum_q |\mathcal{H}_q|}. \tag{12}$$

Thus, it counts contaminated groups, with no five-passage upper bound, and pools group counts across queries. The native counts are 155/380 (2Wiki), 193/297 (MuSiQue), and 342/425 (HotpotQA), yielding the percentages in Figure 4. It is neither a passage-level distractor proportion nor a query-averaged proportion.

Table 7 reports both randomized controls and all three control seeds. Global permutation reassigns passage addresses; degree-preserving swaps randomize group–passage incidence while retaining its degree constraints. The native mean maximum F1 exceeds each randomized control in every dataset. This descriptive comparison shows that group composition carries support-related signal, while the contamination counts show why group membership alone does not define a clean support set. The median fraction of changed groups under the degree-preserving control is 1.0 in each dataset–seed condition.

C.3 Matched propagation check

The propagation check uses 32 training cases per dataset (96 total) and compares the passage scores with an explicit matrix implementation. For incidence matrix H, vertex-degree matrix D_v, edge-size matrix D_e, and diagonal edge weights W, the matched update is $A = D_v^{-1} H W D_e^{-1} H^\top$ on incident vertices, followed by the same restart and passage projection as in Eqs. 7–8. Both unit weights and confidence-derived weights with $\beta = 2$ are checked at $T = 1$ and $T = 2$, with restart $\rho = .35$.

For the two passage-score vectors s and $\widetilde{s}$, agreement requires $\|s - \widetilde{s}\|_\infty \leq 10^{-10} \max(1, \|s\|_\infty, \|\widetilde{s}\|_\infty)$. All 384 comparisons pass, with no empty-seed cases. The maximum absolute and scale-normalized residuals are 1.07×10^{-14} and 7.38×10^{-16}. An independent numerical rerun reproduces the same per-case rows and is not counted as an independent sample. This check concerns algebraic agreement, not retrieval robustness to changing T or the weights.

C.4 SET-LEVEL COMPARISON

The exploratory set-level comparison uses 512 fitting and 128 evaluation cases per dataset from the training split, with BM25 top-10 candidates. It enumerates every non-empty candidate subset of size at most five. The group-completion and pairwise-interaction heads use 33 shared set features, four representation-specific interaction features, and an intercept (38 parameters each). Each is fitted once by query-balanced ridge regression with penalty .01; these are single fitted comparisons rather than replicated training runs. The target utility is

$$U(S, G) = F_{\sup}(S, G) + .5\mathbb{1}[G \subseteq S] - .25\,\frac{|S \setminus G|}{5}. \tag{13}$$

Equal parameter counts specify this comparison without asserting identical functional capacity. Table 8 reports the group-completion and pairwise-interaction heads, pointwise/listwise/MMR selectors at every $k = 1, \ldots, 5$, and two structure-permutation controls. No evaluation-selected best k is substituted for these operating points. The primary group-completion-minus-pairwise contrast uses 10,000 paired query-bootstrap resamples (seed 20260917) and Holm correction across the three datasets; the intervals describe evaluation-query variation for the fitted heads, not training variability.

The primary comparison is group-completion minus pairwise-interaction support F1: -0.004638 (2Wiki), -0.003646 (MuSiQue), and $+0.002307$ (HotpotQA). The corresponding 95% paired-bootstrap intervals are $[-0.015005, 0.003398]$, $[-0.011979, 0.004167]$, and $[-0.010789, 0.015699]$; the Holm-adjusted p-value is .994101 for each contrast. The intervals therefore do not establish a set-level advantage for this parameterization. The table retains every operating point rather than selecting a favorable condition.[1]

Table 8: Set-level selector comparison on BM25 top-10 candidates. Values are evaluation means rounded to four decimal places; $n = 128$ for every row. Group-completion and pairwise-interaction are the two structural heads; pointwise, listwise, and MMR are fixed-cardinality selectors; the two swap conditions permute the group structure at evaluation or before fitting.

Dataset	Selector / condition	F1	Recall	All	Utility	$\lvert S\rvert$	Distractors
2Wiki	pairwise interaction	0.7548	0.9258	0.8281	1.1044	3.4453	1.2891
2Wiki	listwise, $k = 1$	0.5760	0.4238	0.0000	0.5725	1.0000	0.0703
2Wiki	listwise, $k = 2$	0.7435	0.7188	0.4609	0.9533	2.0000	0.4141
2Wiki	listwise, $k = 3$	0.7004	0.8320	0.6328	0.9602	3.0000	1.1328
2Wiki	listwise, $k = 4$	0.6471	0.9102	0.7812	0.9409	4.0000	1.9375
2Wiki	listwise, $k = 5$	0.5774	0.9355	0.8438	0.8563	5.0000	2.8594
2Wiki	MMR, $k = 1$	0.5760	0.4238	0.0000	0.5725	1.0000	0.0703
2Wiki	MMR, $k = 2$	0.7070	0.6816	0.3750	0.8703	2.0000	0.4844
2Wiki	MMR, $k = 3$	0.6826	0.8105	0.5859	0.9166	3.0000	1.1797
2Wiki	MMR, $k = 4$	0.6328	0.8906	0.7266	0.8969	4.0000	1.9844
2Wiki	MMR, $k = 5$	0.5682	0.9238	0.8047	0.8256	5.0000	2.8984
2Wiki	group-completion	0.7501	0.9238	0.8281	1.0982	3.4688	1.3203
2Wiki	evaluation swap	0.6777	0.8672	0.7031	0.9449	3.6953	1.6875
2Wiki	pointwise, $k = 1$	0.5760	0.4238	0.0000	0.5725	1.0000	0.0703
2Wiki	pointwise, $k = 2$	0.7148	0.6895	0.3906	0.8867	2.0000	0.4688
2Wiki	pointwise, $k = 3$	0.6920	0.8223	0.6094	0.9388	3.0000	1.1562
2Wiki	pointwise, $k = 4$	0.6328	0.8906	0.7344	0.9008	4.0000	1.9844
2Wiki	pointwise, $k = 5$	0.5670	0.9238	0.8125	0.8279	5.0000	2.9062
2Wiki	refit swap	0.6990	0.8574	0.6797	0.9670	3.4219	1.4375
MuSiQue	pairwise interaction	0.4863	0.4870	0.1719	0.5125	2.2891	1.1953
MuSiQue	listwise, $k = 1$	0.4617	0.3372	0.0000	0.4492	1.0000	0.2500
MuSiQue	listwise, $k = 2$	0.4919	0.4694	0.1172	0.5033	2.0000	0.9453
MuSiQue	listwise, $k = 3$	0.4698	0.5514	0.2109	0.4878	3.0000	1.7500
MuSiQue	listwise, $k = 4$	0.4294	0.5996	0.2812	0.4380	4.0000	2.6406
MuSiQue	listwise, $k = 5$	0.4018	0.6478	0.3438	0.3975	5.0000	3.5234
MuSiQue	MMR, $k = 1$	0.4247	0.3118	0.0000	0.4087	1.0000	0.3203
MuSiQue	MMR, $k = 2$	0.4740	0.4525	0.0703	0.4599	2.0000	0.9844
MuSiQue	MMR, $k = 3$	0.4708	0.5540	0.2109	0.4888	3.0000	1.7500
MuSiQue	MMR, $k = 4$	0.4354	0.6100	0.3047	0.4565	4.0000	2.6250
MuSiQue	MMR, $k = 5$	0.3943	0.6393	0.3438	0.3885	5.0000	3.5547
MuSiQue	group-completion	0.4827	0.4792	0.1562	0.5014	2.2656	1.1875

[1]Pointwise, listwise, and MMR conditions are evaluated at the displayed fixed cardinality k. The group-completion and pairwise-interaction heads are separate from the main passage-level controller.

Dataset	Selector / condition	F1	Recall	All	Utility	$\lvert S\rvert$	Distractors
MuSiQue	evaluation swap	0.4739	0.4714	0.1406	0.4829	2.2891	1.2266
MuSiQue	pointwise, $k = 1$	0.4247	0.3118	0.0000	0.4087	1.0000	0.3203
MuSiQue	pointwise, $k = 2$	0.4779	0.4564	0.0859	0.4720	2.0000	0.9766
MuSiQue	pointwise, $k = 3$	0.4682	0.5514	0.2109	0.4858	3.0000	1.7578
MuSiQue	pointwise, $k = 4$	0.4420	0.6172	0.3125	0.4681	4.0000	2.6016
MuSiQue	pointwise, $k = 5$	0.3999	0.6458	0.3594	0.4030	5.0000	3.5312
MuSiQue	refit swap	0.4761	0.4688	0.1406	0.4878	2.2266	1.1719
HotpotQA	pairwise interaction	0.6838	0.8203	0.6406	0.9381	2.9609	1.3203
HotpotQA	listwise, $k = 1$	0.5938	0.4453	0.0000	0.5883	1.0000	0.1094
HotpotQA	listwise, $k = 2$	0.7148	0.7148	0.4609	0.9168	2.0000	0.5703
HotpotQA	listwise, $k = 3$	0.6375	0.7969	0.6016	0.8680	3.0000	1.4062
HotpotQA	listwise, $k = 4$	0.5729	0.8594	0.7188	0.8182	4.0000	2.2812
HotpotQA	listwise, $k = 5$	0.5134	0.8984	0.7969	0.7517	5.0000	3.2031
HotpotQA	MMR, $k = 1$	0.5521	0.4141	0.0000	0.5435	1.0000	0.1719
HotpotQA	MMR, $k = 2$	0.7031	0.7031	0.4297	0.8883	2.0000	0.5938
HotpotQA	MMR, $k = 3$	0.6406	0.8008	0.6172	0.8793	3.0000	1.3984
HotpotQA	MMR, $k = 4$	0.5625	0.8438	0.6953	0.7945	4.0000	2.3125
HotpotQA	MMR, $k = 5$	0.5067	0.8867	0.7812	0.7360	5.0000	3.2266
HotpotQA	group-completion	0.6861	0.8242	0.6484	0.9439	2.9766	1.3281
HotpotQA	evaluation swap	0.6768	0.8203	0.6406	0.9280	3.0234	1.3828
HotpotQA	pointwise, $k = 1$	0.5521	0.4141	0.0000	0.5435	1.0000	0.1719
HotpotQA	pointwise, $k = 2$	0.7031	0.7031	0.4375	0.8922	2.0000	0.5938
HotpotQA	pointwise, $k = 3$	0.6375	0.7969	0.6094	0.8719	3.0000	1.4062
HotpotQA	pointwise, $k = 4$	0.5677	0.8516	0.7109	0.8083	4.0000	2.2969
HotpotQA	pointwise, $k = 5$	0.5134	0.8984	0.8047	0.7556	5.0000	3.2031
HotpotQA	refit swap	0.6930	0.8047	0.6094	0.9387	2.7891	1.1797

The structural-swap rows separate evaluation-time perturbation from refitting on perturbed structure. On 2Wiki, native F1 is .750112, versus .677710 with evaluation-time swaps and .699008 after refitting. On HotpotQA, however, refitting on swapped structure yields .693043, above the native .686086. These controls therefore do not establish a universal necessity claim and do not supersede the primary group-completion–pairwise comparison.

D Method Details

D.1 Weighted hypergraph diffusion algorithm

Algorithm 1 formalizes the propagation and projection steps defined in Eqs. 7 and 8. The vertex-to-edge step aggregates vertex scores within each hyperedge, weighted by the fixed confidence-derived edge weight. The edge-to-vertex step distributes hyperedge scores back to vertices with restart. The final projection maps vertex scores to passage scores using tuple factual confidence and hyperedge membership.

D.2 Tuple extraction and weighting

Each passage yields at most five entity-relation-entity tuples. A tuple τ is represented as $(h, r, t, p, c_{\text{fact}}, c_{\text{sal}}, c_{\text{bridge}})$, where h and t are entities, r is the relation, p is the passage ID, and the three confidence scores measure factual likelihood, semantic salience, and bridge potential (as defined in Eq. 1).

Candidate bridge groups collect tuples incident to shared entity v and are retained when they contain at least two tuples and at least three distinct endpoints. For each group, factual confidence and semantic salience are the tuple-wise means, while bridge potential is the tuple-wise maximum. Passage content is never replaced by a generated key: the group only stores the address set P_v.

The default structural signal uses the normalized readout in Eq. 5, followed by the fixed restart propagation depth listed in Table 9. Passage scores add the tuple endpoint term $c_f(\tau)(x_{h_\tau} + x_{t_\tau})$ for tuples in the passage and the group term $w(e)|V_e|^{-1} \sum_{v \in V_e} x_v$ for each group address containing it. The final controller reads structural and text-source ranks, scores, reciprocal ranks, normalized scores, and their interaction features.

The learned baseline controllers use passage-level logistic regression. To avoid overfitting, we use out-of-fold (OOF) predictions during training: each fold's model predicts on held-out training data, which is then used as input to the final ranker. These details define the main passage-level controller and are distinct from the separate set-level comparison.

Algorithm 1 Weighted hypergraph diffusion for passage ranking

Require: Question q, passage set P, hyperedges $\mathcal{E}$, tuples $\mathcal{T}$, restart parameter ρ, steps T
Ensure: Passage scores s_p for $p \in P$

Extract vertices V from tuples $\mathcal{T}$
Initialize seed scores $x_v^{(0)} \leftarrow \text{SeedFromQuestion}(q, V)$ for $v \in V$
Compute vertex degrees $d_v \leftarrow \sum_{e \in \mathcal{E}: v \in V_e} 1$

Vertex-to-edge propagation:
for $t = 1$ to T **do**
    for each hyperedge $e \in \mathcal{E}$ **do**
        $y_e^{(t)} \leftarrow w_e \cdot \frac{1}{|V_e|} \sum_{v \in V_e} x_v^{(t-1)}$
    end for

    Edge-to-vertex propagation with restart:
    for each vertex $v \in V$ **do**
        $x_v^{(t)} \leftarrow \rho \cdot x_v^{(0)} + (1 - \rho) \cdot \frac{1}{d_v} \sum_{e: v \in V_e} y_e^{(t)}$
    end for
end for

Vertex-to-passage projection:
for each passage $p \in P$ **do**
    $s_p \leftarrow \sum_{\tau \in \mathcal{T}_p} c_{\text{fact}}(\tau) \cdot (x_{h_\tau}^{(T)} + x_{t_\tau}^{(T)})$
    $s_p \leftarrow s_p + \sum_{e \in \mathcal{E}_p} \frac{w_e}{|V_e|} \sum_{v \in V_e} x_v^{(T)}$
end for
return $\{s_p : p \in P\}$

The matched propagation check evaluates 96 cases, two weighting choices, and $T \in \{1, 2\}$. It reproduces the implementation with normalized-incidence propagation and the same passage projection, so the paper attributes its contribution to evidence organization and selection rather than to an irreducible propagation operator.

E Data Provenance and Verification

The current support-ID comparisons and diagnostics use fixed data artifacts in the replication package, with aggregate summaries, seed-level summaries, and independent verification records. The extraction cache includes a per-dataset index recording the model, schema, completeness, example counts, and file hashes. Tables of current results transcribe these summaries; the only unit conversion is from fractions to percentages or percentage points.

The independent set-level comparison includes paired-bootstrap intervals and summary hashes. Figure data packets record the source fields and rendering parameters for Figures 3 and 4. Appendix F.3 also summarizes the earlier parameter and depth checks using their original answer-aware metrics.

E.1 Evidence coverage and verification

The main support-ID results are tied to recorded input and output hashes for the cases, candidate views, source scores, selections, and summary files. An independent verifier reports agreement for all 36 aggregate comparisons, 108 comparison–seed summaries, 132 seed-averaged condition summaries, and 268,776 per-case comparisons, with no reported verification errors. These counts describe repeated comparisons over the same queries, not additional independent observations.

The candidate-pool reconstruction covers 22,398 development queries and 2,956,536 selections. Its independent verification checks passage identity and candidate membership and reports no guard failures. It does not recover historical reader-context provenance where that information was not stored; answer-generation performance is therefore outside the evidence used here.

For the set-level comparison, independent verification covers all 384 evaluation cases, 7,296 predictions, 1,920 persisted structure-permuted cases, saved-model reload, summary arithmetic, and paired statistics. Group-composition summaries retain each randomized condition and seed, while the propagation check retains each

numerical residual and its tolerance. These records support the reported results without treating the diagnostics as interchangeable replications.

F Implementation Details

F.1 Tuple Extraction Prompt and Schema

We use a fixed passage-level prompt and JSON schema to extract relation tuples from candidate passages. The extraction is performed once per passage using DeepSeek-V4-Flash with temperature 0 and the resulting cache is reused across all experimental conditions. The schema is part of the fixed input specification: it preserves passage provenance and records three tuple-level scores used by the structural readout.

System: You are an OpenIE triple extraction engine for RAG indexing. Return only valid JSON. Extract compact subject–relation–object triples strictly supported by the supplied passage. Do not answer the question, infer unstated facts, or use benchmark labels, gold supports, or answers.
User: Extract salient relation tuples from the passage below. Use only the supplied passage ID and preserve passage provenance. Return JSON only; do not include Markdown, comments, source spans, high-order relations, or cross-passage relations. Use compact canonical entity names and return an empty list when no supported relation is present.
Passage: {passage_text}
Passage ID: {passage_id}
Output schema:

```
{
  "passages": [
    {"passage_id": "string",
     "triples": [
       {"head": "string", "relation": "string", "tail": "string",
        "factual_confidence": 0.0,
        "semantic_salience": 0.0,
        "bridge_potential": 0.0}
     ]}
  ]
}
```

Rules: Extract at most five salient triples per passage. Scores lie in $[0, 1]$: factual confidence measures textual support, semantic salience measures centrality to the passage, and bridge potential measures whether the tuple can connect evidence across passages. Do not set all scores to the same value when the passage supports a more informative distinction.

The cache uses the top-level key `passages`; each item contains its `passage_id` and a `triples` list with the six extracted fields shown above. The parser attaches the source and passage ID to the internal tuple representation, yielding $(h, r, t, p, c_{\text{fact}}, c_{\text{sal}}, c_{\text{bridge}})$ in Eq. 1. No condition-specific re-extraction is performed.

F.2 Fixed Experimental Settings

Table 9 lists the fixed extraction, structural, and evaluation settings used in the ICLR evidence package. The following subsection explains the shared operating point and summarizes the earlier sensitivity checks.

F.3 Parameter choice and sensitivity checks

Across benchmarks, HKVM uses one shared operating point: $(\lambda_f, \lambda_s, \lambda_b) = (0.20, 0.40, 0.40)$, $\beta = 2.0$, $\rho = 0.35$, and $T = 1$. The normalized 1:2:2 ratio assigns equal weight to semantic salience and bridge potential, with factual confidence at half the coefficient of either field. Since field scores lie in $[0, 1]$, $\beta = 2$ bounds each group multiplier to $[1, 2]$. The restart value $\rho = 0.35$ balances query-seeded activation with group-mediated propagation, followed by a single propagation step. We hold this operating point fixed across benchmarks so the comparisons isolate how relational organization interacts with the evidence and learned selector.

An earlier one-factor screen evaluated 14 settings on the same 2,417 answerable MuSiQue development queries, using seed 13, the cached DeepSeek tuples, and the preceding answer-aware candidate and ranking pipeline. The table reports all-support recall at 10 (AR@10), answer F1, and exact match (EM) for every tested setting.

Table 9: Fixed extraction, structural, and evaluation settings.

Parameter	**Value**
Extraction	
Model	DeepSeek-V4-Flash
Request scope	passage
Passage tuple schema	provenance-preserving six-field tuple record
Maximum tuples per passage	5
Bridge-group construction	
Minimum tuples per group	2
Minimum distinct endpoints	3
Weighted diffusion	
Structural coefficients $(\lambda_f, \lambda_s, \lambda_b)$	$(0.20, 0.40, 0.40)$
Edge-weight scale β	2.00
Restart ρ	0.35
Diffusion steps T	1
Selection and evaluation	
Selector input	structural + text-source features
Passage budget	up to 5 available prefix
Random seeds	13, 29, 43
Primary metrics	support F1, recall, all-support
Seed aggregation	per-case mean, then query-equal dataset mean

Table 10: Earlier one-factor parameter screen on MuSiQue answerable development (seed 13; answer-aware protocol).

Parameter family	Setting	AR@10	Answer F1	EM
Default	$(0.20, 0.40, 0.40),\ \beta = 2.00,\ \rho = 0.35$	37.815	38.946	38.229
β	1.50	37.360	38.693	37.940
	2.50	38.395	39.482	38.767
	3.00	38.395	39.834	39.139
	4.00	39.015	39.918	39.222
ρ	0.15	38.477	39.383	38.726
	0.25	37.857	39.292	38.602
	0.45	37.650	38.937	38.146
	0.55	36.947	38.471	37.650
$(\lambda_f, \lambda_s, \lambda_b)$	$(0.33, 0.33, 0.33)$	37.940	38.988	38.271
	$(1.00, 0.00, 0.00)$	38.188	39.200	38.477
	$(0.00, 1.00, 0.00)$	37.898	38.936	38.229
	$(0.00, 0.00, 1.00)$	37.940	39.022	38.271
	$(0.50, 0.00, 0.50)$	38.022	38.943	38.229

Across the 14 settings, answer F1 ranged from 38.471 to 39.918, a 1.447-point span; the retained default scored 38.946. The tested λ mixtures varied by 0.264 points (38.936–39.200), with the default near the lower end. Higher β and lower ρ increased answer F1, while the best tested alternatives remained within one point of the default. We retained the shared setting as a common readout for comparing evidence organizations, rather than tuning each benchmark to its answer-F1 maximum.

A prior depth comparison showed a mixed pattern (Table 11): T=2 improved 2Wiki by 0.829 points, while changes on MuSiQue and HotpotQA were -0.081 and $+0.054$. We use $T = 1$ as the shared single-step readout across all three benchmarks.

F.4 Bridge-Group Construction Algorithm

Algorithm 2 formalizes the bridge-group construction process described in Section 3.

Groups are formed over the entity endpoints in the candidate tuple cache, without support-label filtering. Query matching initializes the subsequent structural readout; it does not restrict which shared entities can define a group. The direct scan above costs $O(|V|\,|\mathcal{R}|)$; indexing tuples by endpoint avoids repeated scans.

Table 11: Earlier depth comparison under the answer-aware fixed-substrate protocol; values are answer F1.

Dataset	$T = 1$	$T = 2$	$T = 2 - T = 1$
2Wiki	79.299	80.128	+0.829
MuSiQue	39.925	39.843	−0.081
HotpotQA	72.957	73.010	+0.054

Algorithm 2 Bridge-Group Construction

Require: Provenance-preserving candidate tuple cache $\mathcal{R}$
Ensure: Bridge groups $\mathcal{E}$

Initialize $\mathcal{E} \leftarrow \emptyset$
$V \leftarrow \bigcup_{\tau \in \mathcal{R}} \{h_\tau, t_\tau\}$

for each entity $v \in V$ **do**
    // Collect tuples containing v
    $G_v \leftarrow \{\tau \in \mathcal{R} : v \in \{\tau.\text{head}, \tau.\text{tail}\}\}$

    // Count distinct entities in the group
    $V_v \leftarrow \{u : \exists \tau \in G_v, u \in \{\tau.\text{head}, \tau.\text{tail}\}\}$

    // Check bridge criteria
    if $|G_v| \geq 2$ **and** $|V_v| \geq 3$ **then**
        $P_v \leftarrow \{\tau.\text{passage_id} : \tau \in G_v\}$
        $\mathcal{E} \leftarrow \mathcal{E} \cup \{(V_v, P_v)\}$
    end if
end for
return $\mathcal{E}$