Revisiting RAG Retrievers: An Information Theoretic Benchmark

Anonymous ACL submission

Abstract

Retrieval-Augmented Generation (RAG) systems rely critically on the retriever module to surface relevant context for large language models. Although numerous retrievers have recently been proposed, each built on different ranking principles such as lexical matching, dense embeddings, or graph citations, there remains a lack of systematic understanding of how these mechanisms differ and overlap. Existing benchmarks primarily compare entire RAG pipelines or rely on aggregate ranking metrics (Recall, MRR, nDCG) that assume item-level independence, providing little guidance on redundancy or complementarity between retrievers. This work presents MIGRASCOPE, a Mutual Information based RAG Retriever Analysis Scope. We introduce principled metrics grounded in information theory to quantify retrieval quality, redundancy/synergy, and marginal contribution. Although the framework is designed to be applicable to any setting where a ground-truth answer can construct a pseudo chunk distribution, the present empirical study focuses on multi-hop question answering —a demanding setting where retriever complementarity is especially pronounced. On four corpora and thirteen SOTA retrievers spanning lexical, dense, and graph-based families, we show that (i) the MI-based divergence metric captures retriever quality dimensions not reflected by Recall or MRR, (ii) graph-based retrievers are redundant with each other while BM25 remains informationally complementary, and (iii) in the multi-hop QA setting, a carefully chosen ensemble guided by MI-based attribution can outperform any single retriever. We position these findings as task-specific insights and provide a reusable framework for retriever analysis that practitioners can re-estimate on their own target domains.

1 Introduction

Retrieval-Augmented Generation (RAG) has emerged as a dominant paradigm for knowledge-intensive NLP tasks (Lewis et al., 2020). By retrieving relevant external context to inform large language models (LLMs), RAG systems dramatically improve factual accuracy and reduce hallucination (Shuster et al., 2021). The retriever module is the core component responsible for selecting which evidence is passed to the generator, and its effectiveness directly limits downstream performance. A rich ecosystem of retrievers has flourished, including lexical-based methods such as BM25 (Robertson et al., 2009), dense embedding models trained for semantic similarity (Karpukhin et al., 2020), graph-based retrievers designed for multi-hop and global reasoning (Edge et al., 2024), and increasingly hybrid retrieval systems (Sarmah et al., 2024). Each introduces distinct assumptions and ranking mechanisms, yet the field lacks a unified framework to compare them rigorously. Many existing benchmarks focus on new datasets or evaluation metrics applied to full RAG pipelines. While these studies provide valuable insights into overall system quality, they rarely isolate retrieval performance. Some retriever-specific evaluation work has been done, covering heterogeneous domain generalization (Thakur et al., 2021), massive embedding leaderboards (Muennighoff et al., 2023), and complex-objective tasks (Wang et al., 2024b). However these frameworks focus on aggregate ranking-based metrics such as recall, MRR, and nDCG which assume independence between retrieved items and thus fail to explain how different retrievers complement or supersede each other. Recent ensemble-RAG efforts mostly concentrate on hierarchical integration of the full pipeline without disentangling the unique contributions of the retriever itself. As a result, practitioners have limited principled tools for quantifying which retrievers are strong, which are redundant, and which offer complementary value on a given dataset. We argue that revisiting the retriever module with dedicated evaluation tools is essential to unlocking the next

phase of RAG progress.

To address these gaps, we introduce MIGRASCOPE, an information-theoretic framework for analyzing and benchmarking retrievers in RAG. We model each retriever's ranking signal as a noisy view of an underlying chunk probability distribution conditioned on ground-truth answers. Building on Mutual Information (MI), we define a retriever quality score that computes the diverging level from the presumed ideal chunk distribution. This score generalizes across lexical, dense, and graph-based retrieval mechanisms, enabling principled comparison and calibration.

We further develop MI-grounded synergy and redundancy metrics to quantify how multiple retrievers interact - whether they contribute complementary evidence or duplicate signals - and provide visual diagnostics to reveal overlap structure. These tools support task-specific marginal contribution estimation and guide retriever analysis and shortlist construction within a target retrieval setting.

Finally, we translate these analyses into practice with a lightweight ensemble that uses MI-based attribution to select and weight retrievers with minimal supervision. Applied across major RAG corpora and a broad suite of state-of-the-art (SOTA) retrievers, the framework yields new insights into contribution patterns, redundancy, and robustness, and the ensemble consistently outperforms strong single retrievers in our evaluated multi-hop QA setting. This paper makes the following contributions:

- Introduce MIGRASCOPE, an MI based retriever quality score that normalizes across ranking paradigms (lexical, dense, graph) to quantify retrieval relevance and utility.
- Define and visualize MI-grounded synergy and redundancy metrics for multiple retrievers, revealing complementary and overlapping signal structure within ensembles.
- Develop a retriever ensemble framework with MI-based attribution to estimate each retriever's marginal contribution within a task-specific retrieval setting.
- Benchmark a broad set of graph-RAG and standard RAG retrievers using the proposed methodology, yielding new insights on contribution patterns, robustness, and design guidance for multi-hop QA RAG systems.

Our results suggest a need to shift community focus from endlessly inventing new retrievers toward understanding how existing ones differ, overlap, and cooperate within a target retrieval environment. Through an information-theoretic lens, we uncover structure and simplicity hidden beneath the diversity of modern retrieval methods, paving the way for more reliable and interpretable RAG systems.

2 Related Work

Our work is positioned at the intersection of three active research areas: RAG benchmarking, RAG ensemble methods, and the application of information-theoretic principles to large language models.

2.1 RAG Benchmarking

Recent work on RAG evaluation has primarily focused on establishing new datasets, defining system-level metrics, or analyzing operational costs, rather than systematically comparing the retrievers themselves. Several benchmarks have been introduced to test RAG systems on complex tasks, such as multi-hop reasoning over structured documents (Khang et al., 2025) or domain-specific GraphRAG applications (Xiao et al., 2025). Others have focused on creating datasets for specific industrial domains, such as finance (Wang et al., 2024a), or on proposing new evaluation methodologies, like assessing LLM robustness and information integration (Chen et al., 2024b) or developing explainable metrics for industrial settings (Friel et al., 2024).

To evaluate retrievers in isolation, the Information Retrieval community has developed rigorous benchmarks covering heterogeneous domain generalization (Thakur et al., 2021), massive embedding leaderboards (Muennighoff et al., 2023), and complex user objectives (Wang et al., 2024b). However, these frameworks typically rely on aggregate ranking metrics such as nDCG and Recall. While effective for static leaderboards, these metrics treat document utility as additive, implicitly assuming independence between retrieved items. Consequently, they fail to quantify redundancy or measure the conditional information gain required to understand how distinct retrievers complement or supersede one another in an ensemble.

While valuable, these contributions often treat the retriever as a component within a larger pipeline. The focus remains on dataset design or system-level assessment. Other studies analyze important, but orthogonal, aspects. For instance, Lin (2024) provide practical guidance on the operational trade-offs of different retrieval infrastructures (e.g., HNSW vs. inverted indexing) but do not offer prescriptive insights on retriever algorithm se-

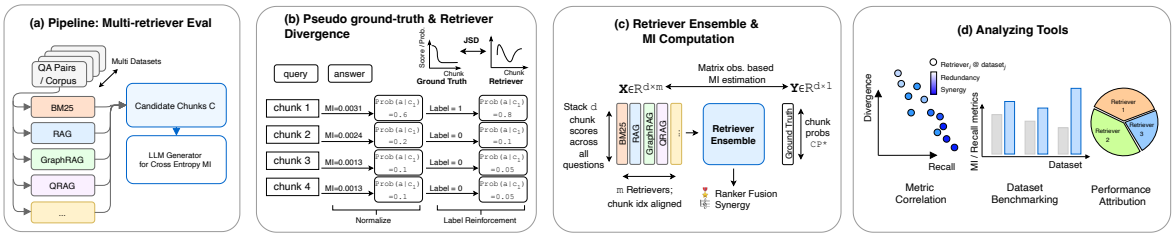

Figure A: The proposed MIGRASCOPE analyzing framework

lection. Similarly, Zhang et al. (2025a) compare different LLMs as dense retrievers, but limit their analysis to this single family.

Across these studies, there remains a significant gap in understanding the fundamental differences, redundancies, and complementary strengths of the various retriever families (e.g., dense, sparse, hybrid, and graph-based). Our work directly addresses this gap by proposing a framework to benchmark the retrievers themselves.

2.2 RAG Ensemble Methods

As RAG systems have matured, ensemble and multi-retriever architectures have become increasingly common. Many of these efforts focus on sophisticated pipeline orchestration. For example, Zhang et al. (2025b) propose a hierarchical system with a planner that orchestrates multiple low-level searchers, while Wu et al. (2025) introduce a modular framework that composes distinct steps like question decomposition and retrieval. Other approaches integrate heterogeneous data sources, such as combining web search with knowledge graphs (Xie et al., 2024) or creating adaptive hybrid systems for specific domains like law (Kalra et al., 2024).

A separate line of work seeks to optimize the information passed to the generator. This includes methods for decoupling retrieval and generation representations (Yang et al., 2025) or explicitly balancing relevance and semantic diversity in the retrieved document set (Rezaei and Dieng, 2025). Chen et al. (2025) explicitly analyze ensembles through an information-theoretic lens, arguing that they increase the total useful information. However, their work stops short of providing an operational recipe for selecting or weighting retrievers. These studies primarily offer architectural solutions or in-domain systems, but do not provide a principled, data-driven method for understanding *which* retrievers to ensemble or *how* to combine them based on their quantified marginal contribution.

2.3 Information-Theoretic Analysis in RAG

Information theory provides a powerful, formal toolkit for quantifying concepts like uncertainty, relevance, and redundancy, which are central to RAG. Recently, these tools have been applied to various parts of the RAG pipeline. For instance, Zhu et al. (2024) use an information bottleneck formulation to filter noise from retrieved contexts. Liu et al. (2024) propose using Pointwise Mutual Information (PMI) as a probabilistic gauge for RAG performance, while others have used divergence metrics to measure attribution (Li et al., 2025) or detect hallucinations (Halperin, 2025). These methods successfully apply information-theoretic concepts to *diagnose* or *optimize* a specific RAG component, such as context pruning (Deng et al., 2025) or detecting data memorization (Anh et al., 2025).

However, this body of work has not yet been directed at the problem of *retriever evaluation*. While Yumuşak (2025) outline a theoretical framework for RAG information flow, it lacks the empirical benchmarking to reveal how different SOTA retrievers actually perform under these metrics. To our knowledge, no existing work uses mutual information and information divergence to create a comparative framework that quantifies the quality, redundancy, and marginal contribution of distinct retriever families.

In summary, the field currently lacks a unified framework for comparing retriever modules directly, a principled method for designing retriever ensembles based on redundancy, and an empirical application of information theory to benchmark retriever contributions. Our work, MIGRASCOPE, is designed to fill these precise gaps.

3 Analytical Framework

3.1 Preliminaries: GraphRAG Formulations

Let a query be $q \in \mathcal{Q}$, a candidate chunk set be $C = \{c_1, \ldots, c_N\}$, and an answer be $a \in \mathcal{A}$. A retriever r maps q to a scored or ranked subset $\mathcal{C}_q \subseteq C$ as the context. The indices of chunks

subset $\mathcal{C}_q$ are denoted as $\mathcal{I}_q$. The retriever gives each chunk a score, combining into a scored list $\{(c_j, s_j)\}_{j \in \mathcal{I}_q}$. A generator then produces the answer a conditioned on $(q, \mathcal{C}_q)$. We formulate a simplified version of RAG and GraphRAGs here. More details are described in Appendix A.

Vanilla RAG. The retrieval $\mathcal{C}_q$ relies solely on vector similarity to select the top-k chunks:

$$\mathcal{C}_q = \mathrm{Top}_k(\{c_j\}_{j=1}^N, \mathbf{Sim}(\mathbf{Embed}(q), \mathbf{Embed}(c_j))) \tag{1}$$

GraphRAG (General Form). GraphRAG introduces a knowledge graph G constructed from the corpus. Retrieval is a two-stage process: graph anchoring and context mapping.

$$G = \mathbf{Construct}(C) \tag{2}$$

$$\mathcal{C}_q = \mathbf{Map}_{G \to C}(\mathbf{Anchor}(G, q)) \tag{3}$$

3.2 Information-Theoretic View of RAG

Mutual information (MI). For random variables X, Y with joint $p(x, y)$ and marginals $p(x), p(y)$,

$$\begin{aligned} I(X;Y) &= \mathbb{E}_{p(x,y)}\left[\log \frac{p(x,y)}{p(x)\,p(y)}\right] \\ &= H(X) - H(X \mid Y) = H(Y) - H(Y \mid X). \end{aligned} \tag{4}$$

MI measures the reduction in uncertainty of one variable given the other; it is zero iff X and Y are independent.

Divergence. In this work we apply Jensen–Shannon divergence (JSD) which is symmetric, bounded in $[0, \log 2]$, and well-defined even with disjoint supports.

$$\begin{aligned} \mathrm{KL}(P \parallel Q) &= \mathbb{E}_p\left[\log \frac{p}{q}\right], \\ \mathrm{JSD}(P, Q) &= \tfrac{1}{2}\,\mathrm{KL}\big(P \| M\big) + \tfrac{1}{2}\,\mathrm{KL}\big(Q \| M\big), \\ M &= \tfrac{1}{2}(P + Q). \end{aligned} \tag{5}$$

Pointwise mutual information in RAG. Given query q and a random multiset of retrieved chunks $\mathcal{C}_q$, define the pointwise mutual information w.r.t. the answer a as

$$\mathrm{PMI}(a; \mathcal{C}_q \mid q) = \log \frac{p(a \mid \mathcal{C}_q, q)}{p(a \mid q)}. \tag{6}$$

Exact evaluation is intractable because the answer a is generative with a combinatorially large support. In practice we approximate with a fixed ground-truth answer a from the dataset:

$$\widehat{\mathrm{PMI}}(a; \mathcal{C}_q \mid q) \approx \log p(a \mid \mathcal{C}_q, q) - \log p(a \mid q). \tag{7}$$

We estimate $\log p(a \mid \cdot)$ with the LLM token-level cross-entropy (log perplexity). For a tokenization $a = (t_1, \ldots, t_T)$ and model condition Z,

$$\begin{aligned} &\log p(a \mid Z) = \sum_{t=1}^{T} \log p_\theta(t_t \mid t_{<t}, Z) \\ &\iff \quad \mathrm{CE}(a \mid Z) = -\frac{1}{T}\sum_{t=1}^{T} \log p_\theta(t_t \mid t_{<t}, Z). \end{aligned} \tag{8}$$

where θ represents the generative LLM parameters. Large PMI between the answer a and context $\mathcal{C}_q$ corresponds to large perplexity drop under retrieved context $Z = (q, \mathcal{C}_q)$ relative to no context $Z = (q)$. PMI, then, quantifies how much a retrieved set of chunks C_q, for a given question q, reduces uncertainty in the ground-truth answer a.

3.3 Pseudo Ground-Truth Chunk Probability

We define a pseudo ground truth probability over the retrieved chunks that reflects how well each chunk supports the known answer. Related chunk attribution techniques utilize LLM cross-entropy to gauge chunk importance, either by calculating Information Bottleneck scores (Zhu et al., 2024) or by measuring the performance degradation when a specific chunk is removed from the context (Deng et al., 2025). However, such methods often introduce undesirable complexity. Ablation-based approaches suffer from computational overhead and bias induced by chunk order, as LLMs have been shown to disproportionately favor the first and last chunks in a sequence (Liu et al., 2024). Meanwhile, Information Bottleneck methods require tuning sensitive hyperparameters and optimizing auxiliary reconstruction objectives. To avoid these pitfalls, we apply a simple, independent scoring approach to measure the chunk self-attribution score. Thus we have the pseudo ground-truth chunk probability CP^*:

$$\mathrm{CP}^*(c \mid q, a) \;= \mathrm{Softmax}_{\mathcal{C}_q}(\log p_\theta(a \mid q, c)) \tag{9}$$

where $\mathrm{Softmax}_{\mathcal{C}_q}$ normalizes the chunk probabilities based on the logits over the subset of retrieved chunks $\mathcal{C}_q$.

Golden chunk reinforcement. If the dataset provides golden supporting chunks $G_q \subseteq \mathcal{I}_q$, we optionally reinforce them by a scalar $\gamma > 1$, i.e. amplify the dataset labeled ground truth chunks then re-normalize [1]:

$$\widetilde{\mathrm{CP}}^*(c \mid q, a) = \frac{\mathbf{1}[c \in G_q]\gamma \cdot \mathrm{CP}^*(c \mid q, a) + \mathbf{1}[c \notin G_q] \cdot \mathrm{CP}^*(c \mid q, a)}{\sum_{j \in \mathcal{I}_q} (\mathbf{1}[c_j \in G_q]\gamma + \mathbf{1}[c_j \notin G_q])\mathrm{CP}^*(c_j \mid q, a)}. \tag{10}$$

We use $\widetilde{\mathrm{CP}}^*$ as the default pseudo target distribution to produce the following retriever analysis.

3.4 Retriever Divergence Score

Once we have the pseudo ground-truth chunk probability CP^* as the standard to compare with, we can easily measure the quality of any retriever r by comparing the scores produced by this retriever and CP^*. We first convert the retriever scores $s_r(c)$ to probabilities:

$$p_r(c \mid q) = \frac{\exp(s_r(c)/\tau)}{\sum_{j \in \mathcal{I}_q} \exp(s_r(c_j)/\tau)} \tag{11}$$

where $\tau > 0$ is a temperature-softmax normalization. The quality of the retriever r can be defined by the divergence of r from the pseudo target:

$$\mathrm{Div}(r) \;=\; \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \mathrm{JSD}\Big(p_r(\cdot \mid q) \;\Big\|\; \widetilde{\mathrm{CP}}^*(\cdot \mid q, a_q)\Big). \tag{12}$$

3.5 Ensembling Multiple Retrievers

Assume m retrievers $\{r_i\}_{i=1}^m$ produce scores $\{s_i(c)\}_{i=1}^m$. There are multiple ways that these retrievers can be fused. We list and compare 12 ways to merge the rankers and show results in Section 4. We do not aim to claim novelty on the ranker fusion approaches; we leverage existing approaches as detailed in Appendix B.1.

3.6 Marginal Contributions and Redundancy Analysis

In the attempt to benchmark and analyze different retrievers, we utilize well-established tools from statistical signal processing to quantify the contribution and redundancy levels.

Setup. Given a batch of queries $\{q_i\}_{i=1}^{|\mathcal{Q}|}$, we stack the retriever's scores and CP^* to obtain observation X and estimation target Y.

For each query, we use $\widetilde{\mathrm{CP}}^*$ to produce a list of top K chunk scores, and concatenate them into target vector $Y \in \mathbb{R}^{K|\mathcal{Q}|}$

[1] When golden chunk support exists, the chunk distribution is ideally one-hot. The CP^* softens the hard distribution. For better alignment with dataset labels, we empirically pick large γ.

On the other hand, each of the m retrievers also produces scores for these top K chunks. We stack them into the observation matrix $X \in \mathbb{R}^{K|\mathcal{Q}| \times m}$.

Utility. Define the set function

$$F(S) = I\big(Y; X_S\big), \tag{13}$$

which measures predictive information about Y contained in the subset S of retrievers. This is distinct from Fisher information.

Independent marginal contribution. Denote the collection of m retrievers as $\mathcal{R}$. For retriever i, its independent marginal contribution is:

$$\begin{aligned} C_i \;&\stackrel{\text{def}}{=}\; I\big(Y; X_i \mid X_{\mathcal{R}/\{i\}}\big) \\ &=\; H\big(Y \mid X_{\mathcal{R}/\{i\}}\big) - H\big(Y \mid X_{\mathcal{R}}\big), \end{aligned} \tag{14}$$

where $X_{\mathcal{R}/\{i\}}$ denotes all columns of X excluding the column corresponding to retriever i, $X_{\mathcal{R}}$ denotes the full observation matrix. Large C_i then captures the unique information conveyed by retriever i. Values near zero indicate redundancy given the others.

Shapley value for fair attribution. Define Shapley values over the coalition game $(\{1, \ldots, m\}, F)$:

$$\phi_i = \sum_{S \subseteq \{1,\ldots,m\} \setminus \{i\}} \frac{|S|!\,(m - |S| - 1)!}{m!} \left[F(S \cup \{i\}) - F(S)\right]. \tag{15}$$

Shapley values fairly allocate the total utility $F(\{1{:}m\}) - F(\varnothing)$ across retrievers by $\Sigma_{i=1}^m \phi_i = F(\{1{:}m\}) - F(\varnothing)$. They average marginal information gains over all insertion orders, which resolves order dependence.

Redundancy vs synergy. For any pair (i, j), define the interaction information

$$\mathrm{II}(Y; X_i; X_j) \;=\; I(Y; X_i) - I\big(Y; X_i \mid X_j\big). \tag{16}$$

Positive II indicates redundancy; negative II indicates synergy, i.e., complementary information that emerges only jointly. Accordingly, we define a surrogate distance between retrievers r_i and r_j as

$$d(r_i, r_j) = \exp\big(-\mathrm{II}(Y; X_i; X_j)\big). \tag{17}$$

We further leverage this score to position each retriever on a *redundancy–synergy spectrum*, and perform clustering analysis to reveal the structural grouping of retrievers. Results are presented in Section 4.6.

Retriever Selection Principle. We apply the tractable redundancy/synergy scores in Eq. (16) to decide whether to include certain RAG retriever for ensemble. We follow the principle that multi-collinearity harms estimation stability but independence helps. For any sources X_1, X_2 and target Y, $I(Y; X_1, X_2) = I(Y; X_1) + I(Y; X_2|X_1)$. If X_2 is redundant given X_1, then $I(Y; X_2|X_1) = 0$. If X_2 is complementary, this conditional term is positive, so the pair contains strictly more information than either alone.

3.7 Estimating Entropy and Mutual Information from Vector Observations

Assuming X and Y are jointly Gaussian yields a closed-form mutual information (MI) estimator (see Appendix D.1). However, this assumption is theoretically tenuous in our setting and, empirically, leads to systematic underestimation of MI by failing to capture nonlinear dependencies between the variables.

Therefore, we accept the distributions of X and Y as unknown, then train a predictive model $\hat{p}(y \mid x)$ and compute $\widehat{H}(Y \mid X) = \frac{1}{n}\sum_i H(\hat{p}(\cdot \mid x_i))$ for discrete Y, or approximate continuous Y with a Gaussian residual model where $\widehat{H}(Y \mid X) = \frac{1}{2}\log(2\pi e\, \widehat{\sigma}^2_{\text{res}})$. We use XGB for regression.

4 Results

4.1 Benchmarking RAG Retrievers

In this work, we focus on the effect of retriever mechanisms rather than encoder, hence the encoder remains fixed and same for all RAG approaches in our work. We use BGE-M3 (Chen et al., 2024a) indiscriminatively. We benchmark a dozen of SOTA RAG retrievers, including chunk similarity based (vanilla) retriever, lexical retrievers, graph retrievers, and chunk decomposition based retrievers, etc. For the seven G-series graph retrievers, the encoder and indexing backbone are shared, while the graph construction differs: windowed co-occurrence, thresholded co-occurrence, hierarchical provenance, naive co-occurrence, community, relationship-global, and global co-occurrence. We additionally compare with HippoRAG (Gutiérrez et al., 2025) and LightRAG (Guo et al., 2024).

4.2 Datasets and Split

We evaluate the retrievers on four multi-hop QA corpora: HotpotQA, MuSiQue, 2WikiMultiHopQA, TriviaQA. For each corpus, we uniformly sample 1,000 QA pairs.

Benchmarking results for these retrievers are presented in Table 1, which shows the traditional metrics (Recall@1, MRR) as well as the newly derived divergence metric from Eq. (12). Next, we leverage the proposed tools to address four key research questions.

- RQ1: How does the new MI based metric align with previous MRR/Recall metrics?
- RQ2: How does the selection of hyper-parameters impact MI?
- RQ3: Is it possible to combine multiple weak retrievers into a stronger retriever, and how are they contributing individually?
- RQ4: Among the SOTA retrievers, how redundant are they?

More results are presented in Appendix C, including variances, computational costs, ranking robustness, case study of MI/Recall metric differences, MIGRASCOPE analyzing workflow, etc.

4.3 Correlation Analysis (RQ1)

This experiment aims to check the level of alignment between the proposed divergence metric and traditional metrics. For each (retriever, dataset) pair, we plot the divergence score against standard evaluation metrics (Recall@1, MRR) in Figure 1.

The first observation is that recall and *Div* scores are negatively correlated in general (strong retrievers tend to have lower divergence). There are also systematic deviations, as multiple retrievers achieve similar recall but differ significantly in divergence. This confirms the complementary nature of our divergence metric and its ability to diagnose failure modes not captured by ranking accuracy alone.

These deviations also affect practical selection decisions. On 2Wiki, rank-based metrics among strong graph retrievers can be contradictory, while *Div* selects the retriever that later contributes more strongly in the best ensemble. On MuSiQue, several graph variants are marginally stronger than G-hierarchical under Recall/MRR, yet *Div* identifies G-hierarchical as the better ensemble component.

4.4 Sensitivity of MI Hyperparameters (RQ2)

We sweep top-K, anchor construction, and golden-chunk reinforcement strength γ; the full plot is deferred to Appendix C.3 to conserve space. The

Method	2Wiki			HotpotQA			MuSiQue			TriviaQA		
	Rec	MRR	Div	Rec	MRR	Div	Rec	MRR	Div	Rec	MRR	Div
G-window	0.935	0.959	0.110	0.871	0.916	0.118	0.822	0.806	0.125	0.891	0.922	0.102
G-threshold	0.936	0.960	0.120	0.921	0.924	0.121	0.725	0.805	0.122	0.944	0.922	0.101
G-hierarchical	0.934	0.958	0.117	0.872	0.916	0.120	0.724	0.803	0.129	0.881	0.914	0.098
G-naive	0.957	0.934	0.122	0.876	0.920	0.125	0.729	0.808	0.134	0.892	0.933	0.144
G-community	0.935	0.959	0.125	0.873	0.917	0.123	0.727	0.807	0.130	0.891	0.919	0.119
G-relationship	0.934	0.958	0.128	0.873	0.917	0.124	0.728	0.807	0.129	0.845	0.922	0.127
G-global	0.937	0.961	0.125	0.880	0.922	0.128	0.726	0.805	0.131	0.835	0.955	0.121
HippoRAG	0.931	0.960	0.142	0.892	0.921	0.128	0.833	0.810	0.120	0.921	0.920	0.119
RAG	0.911	0.921	0.133	0.851	0.889	0.130	0.711	0.768	0.159	0.860	0.891	0.125
LLM	0.913	0.944	0.131	0.824	0.908	0.138	0.807	0.801	0.155	0.882	0.914	0.129
QRAG	0.889	0.919	0.147	0.713	0.792	0.133	0.681	0.753	0.166	0.829	0.881	0.144
LightRAG	0.785	0.863	0.155	0.507	0.644	0.162	0.563	0.659	0.199	0.877	0.820	0.157
BM25	0.486	0.605	0.252	0.620	0.739	0.238	0.415	0.507	0.302	0.420	0.569	0.289

Table 1: RAG Retriever Benchmarking with MI metric

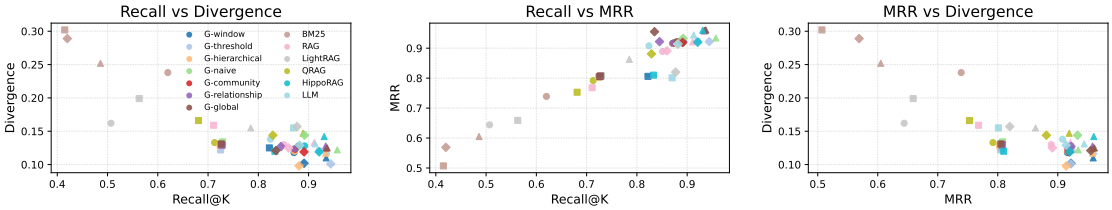

Figure 1: Metric Correlations Across Datasets, Legend by Retriever

same qualitative pattern remains: larger γ makes *Div* more recall-like, while larger top-K reduces Pearson alignment because retrievers increasingly differ from the pseudo target away from the top evidence. Importantly, retriever ordering is more stable than raw Pearson values alone suggest. On HotpotQA, the Spearman correlation of *Div*-induced retriever rankings across the sweep stays within $[0.748, 0.910]$, with per-setting means between 0.841 and 0.852 (Table 4). This supports using *Div* primarily as a relative diagnostic for model selection rather than an absolute standalone score.

4.5 Retriever Ensemble and Shapley Contribution Values (RQ3)

We conduct a series of experiments in the attempt to find a subset of combined retrievers outperforming any single strongest retriever, as well as visualizing the synergy across different retrievers.

Our retriever ensemble is based on the divergence metric. For every given dataset, we first run retrieval procedure with every retriever to get a large top-K chunk list for 1000 QA pairs. Then we select a list of retriever subset to perform ensemble search in the unified search space. The best fusion rule is selected on the **20% training split** by maximizing held-out MI on a nested 5-fold cross-validation within that split; no test labels are touched during fusion-rule selection. The ensembled retriever is then evaluated on the **remaining 80% test QA pairs**. Operationally, this also yields a practitioner-facing shortlist procedure: score retrievers individually by *Div*, remove dominated or highly redundant candidates, add retrievers greedily by marginal MI/Shapley gain, and tune only a small family of fusion rules on the resulting shortlist (Appendix C.5).

We break down this research question into several key points for a better comprehensive understanding. **RQ3.1** How much are these retrievers contributing to the ensembled results? **RQ3.2** How much information does the best ensemble strategy provide? What if it deviates from the best configuration? **RQ3.3** Ensemble stability: if more retrievers are added to the best ensemble configuration, or some retrievers are dropped, how does it impact the performance?

4.5.1 Best Recall by Shapley Shares (RQ3.1)

We compute the Shapley attribution values for every retriever within the best retriever set in Figure 3.1. The height of the bar plot means the overall recall performance relative to 1.0. From this figure we can see that the strongest retriever also tends to contribute the most. However, the winner is not the ensemble of the top-performing GraphRAG retrievers, but it often includes worse retrievers such as BM25. This hints that different GraphRAG retriev-

ers may contain redundant components that harm synergy.

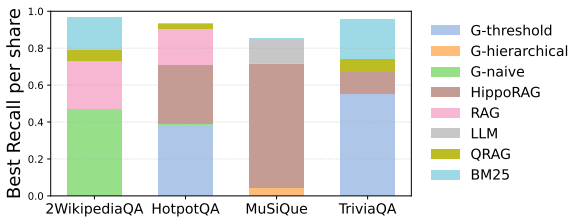

Figure 3.1: Shapley share for the best ensemble configs. Variances see Appendix C.1.

4.5.2 Perturbation Impact on MI (RQ3.2)

We evaluate RQ3.2 by computing $I(\widetilde{\mathrm{CP}}^*; X_S)$ from Eq. 13 for the best retriever subset and its perturbed variations, where S is some retriever ensemble set. The results are plotted in Figure 3.3. In this figure, the y-axis labels are different retriever subsets, ranked by the MI. As can be seen in the figure, the few top performing retriever subsets show similar performance. However, when the perturbation of retrievers are too large, the performance drops drastically.

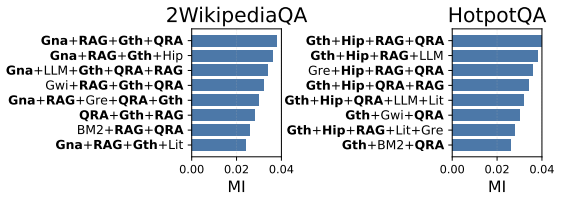

Figure 3.3: MI under ensemble perturbation.

4.5.3 Perturbation Impact on Recall (RQ3.3)

We answer RQ3.3 by first identifying the best retriever subset for ensemble, then gradually add or drop retrievers on top of this best subset, and finally concatenate all results into a continuous line. We plot the results in Figure 3.2.

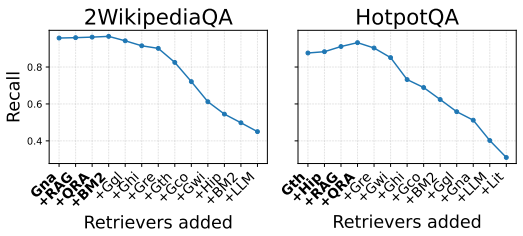

Figure 3.2: Recall under ensemble perturbation.

As can be seen in Figure 3.2, if the retriever subset is chosen wisely, it performs better than single best retriever. As more retrievers are added, redundancy dominates over synergy, hence the performance dropped drastically.

4.6 Retriever Redundancy Spectrum (RQ4)

We formulate the visualization of the m retrievers as a geometric embedding problem. We first pre-compute the pairwise redundancy for all retrievers. By interpreting the pairwise redundancy as a measure of similarity, we derive a distance metric where highly redundant pairs correspond to short distances as in Eq. 17. Using Classical Multidimensional Scaling (MDS), we map the high-dimensional relationship matrix into a 2D Euclidean space. This mapping preserves the global relational structure, allowing us to visualize retriever clusters based on their informational synergy, as shown in Figure 4.

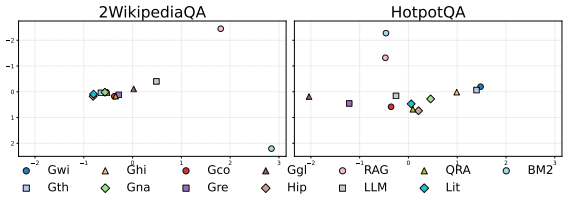

Figure 4: Retriever redundancy/synergy spectrum.

As can be observed in the figure, GraphRAG approaches are similar to each other and further away from vanilla RAG or lexical BM25 retriever. This implies that redundancy might be clustered by approach types (graph vs. lexical vs. vanilla).

5 Conclusion

We introduced MIGRASCOPE, an information-theoretic framework for evaluating RAG retrievers with principled metrics for quality, redundancy, and marginal contribution. By modeling rankings as noisy views of an answer-conditioned chunk distribution, it provides a task-specific way to measure retriever quality, redundancy, and marginal utility, and shows on multi-hop QA benchmarks that well-designed ensembles can outperform individual retrievers. Despite current limitations, MIGRASCOPE offers new insight into retriever interactions and practical tools for building more robust and efficient RAG systems. Future work should integrate generation-side metrics, study retriever-specific internal mechanisms, improve computational efficiency for large-scale deployment, and evaluate the framework beyond multi-hop QA to test how redundancy and complementarity vary across domains.

Limitations

As a pioneering effort in applying mutual information to retriever evaluation, MIGRASCOPE has certain limitations that suggest opportunities for refinement. The MI-based metrics, while theoretically grounded, are less familiar than conventional metrics like Recall@K or MRR, and their interpretation may require additional context for practitioners. The framework depends on hyperparameters and estimator choices, which can influence numerical stability and relative rankings, though sensitivity analysis can mitigate these effects. Additionally, the reported MI values are approximations influenced by model assumptions, estimator errors, and practical truncations, and should be viewed as consistent estimates under specific settings. The present instantiation is also chunk-local: $\widetilde{\mathrm{CP}}^*$ scores each chunk independently, so it does not explicitly model evidence that is useful only jointly as a chunk pair or set. Finally, our empirical study is limited to multi-hop QA, so the redundancy and complementarity structures reported here should be interpreted as task-specific and re-estimated for new domains rather than assumed to transfer unchanged; accordingly, the current results should not be read as universal retriever-selection rules across RAG settings.

Additionally, we do not compare against ensemble-diversity metrics from the classifier ensemble literature (e.g., Q-statistic, disagreement measure) because these are defined for binary prediction settings and do not generalize to graded ranked retrieval lists; the conceptual mismatch would make any such comparison misleading. Similarly, while we evaluate 12 ranker fusion strategies and identify the best per dataset, we do not claim novelty over any individual fusion rule. The contribution is the *selection criterion* (MI-based attribution) rather than the fusion mechanism itself. The standard RRF baseline (Cormack et al., 2009) is included among the 12 strategies and consistently underperforms the MI-guided weighted fusion, supporting the value of principled retriever selection over naïve rank aggregation.

References

Le Vu Anh, Nguyen Viet Anh, Mehmet Dik, and Luong Van Nghia. 2025. Repcs: Diagnosing data memorization in llm-powered retrieval-augmented generation. *arXiv preprint arXiv:2506.15513*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2024b. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17754–17762.

Yifei Chen, Guanting Dong, Yutao Zhu, and Zhicheng Dou. 2025. Revisiting rag ensemble: A theoretical and mechanistic analysis of multi-rag system collaboration. *arXiv preprint arXiv:2508.13828*.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Jiale Deng, Yanyan Shen, Ziyuan Pei, Youmin Chen, and Linpeng Huang. 2025. Influence guided context selection for effective retrieval-augmented generation. *arXiv preprint arXiv:2509.21359*.

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

Robert Friel, Masha Belyi, and Atindriyo Sanyal. 2024. Ragbench: Explainable benchmark for retrieval-augmented generation systems. *arXiv preprint arXiv:2407.11005*.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024. Lightrag: Simple and fast retrieval-augmented generation. *arXiv preprint arXiv:2410.05779*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From rag to memory: Non-parametric continual learning for large language models. *arXiv preprint arXiv:2502.14802*.

Igor Halperin. 2025. Prompt-response semantic divergence metrics for faithfulness hallucination and misalignment detection in large language models. *arXiv preprint arXiv:2508.10192*.

Rishi Kalra, Zekun Wu, Ayesha Gulley, Airlie Hilliard, Xin Guan, Adriano Koshiyama, and Philip Treleaven. 2024. Hypa-rag: A hybrid parameter adaptive retrieval-augmented generation system for ai legal and policy applications. *arXiv preprint arXiv:2409.09046*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi

Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Minsoo Khang, Sangjun Park, Teakgyu Hong, and Dawoon Jung. 2025. Crest: A comprehensive benchmark for retrieval-augmented generation with complex reasoning over structured documents. *arXiv preprint arXiv:2505.17503*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Ruizhe Li, Chen Chen, Yuchen Hu, Yanjun Gao, Xi Wang, and Emine Yilmaz. 2025. Attributing response to context: A jensen-shannon divergence driven mechanistic study of context attribution in retrieval-augmented generation. *arXiv preprint arXiv:2505.16415*.

Jimmy Lin. 2024. Operational advice for dense and sparse retrievers: Hnsw, flat, or inverted indexes? *arXiv preprint arXiv:2409.06464*.

Tianyu Liu, Jirui Qi, Paul He, Arianna Bisazza, Mrinmaya Sachan, and Ryan Cotterell. 2024. Pointwise mutual information as a performance gauge for retrieval-augmented generation. *arXiv preprint arXiv:2411.07773*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037.

Mohammad Reza Rezaei and Adji Bousso Dieng. 2025. Vendi-rag: Adaptively trading-off diversity and quality significantly improves retrieval augmented generation with llms. *arXiv preprint arXiv:2502.11228*.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 608–616.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. *arXiv preprint arXiv:2104.07567*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Shuting Wang, Jiejun Tan, Zhicheng Dou, and Ji-Rong Wen. 2024a. Omnieval: An omnidirectional and automatic rag evaluation benchmark in financial domain. *arXiv preprint arXiv:2412.13018*.

Xiaoyue Wang, Jianyou Wang, Weili Cao, Kaicheng Wang, Ramamohan Paturi, and Leon Bergen. 2024b. Birco: A benchmark of information retrieval tasks with complex objectives. *arXiv preprint arXiv:2402.14151*.

Ruofan Wu, Youngwon Lee, Fan Shu, Danmei Xu, Seung-won Hwang, Zhewei Yao, Yuxiong He, and Feng Yan. 2025. Composerag: A modular and composable rag for corpus-grounded multi-hop question answering. *arXiv preprint arXiv:2506.00232*.

Yilin Xiao, Junnan Dong, Chuang Zhou, Su Dong, Qianwen Zhang, Di Yin, Xing Sun, and Xiao Huang. 2025. Graphrag-bench: Challenging domain-specific reasoning for evaluating graph retrieval-augmented generation. *arXiv preprint arXiv:2506.02404*.

Weijian Xie, Xuefeng Liang, Yuhui Liu, Kaihua Ni, Hong Cheng, and Zetian Hu. 2024. Weknow-rag: An adaptive approach for retrieval-augmented generation integrating web search and knowledge graphs. *arXiv preprint arXiv:2408.07611*.

Peiru Yang, Xintian Li, Zhiyang Hu, Jiapeng Wang, Jinhua Yin, Huili Wang, Lizhi He, Shuai Yang, Shangguang Wang, Yongfeng Huang, and 1 others. 2025. Heterag: A heterogeneous retrieval-augmented generation framework with decoupled knowledge representations. *arXiv preprint arXiv:2504.10529*.

Semih Yumuşak. 2025. An information-theoretic framework for retrieval-augmented generation systems. *Electronics*, 14(15):2925.

Hengran Zhang, Keping Bi, and Jiafeng Guo. 2025a. A comparative study of specialized llms as dense retrievers. *arXiv preprint arXiv:2507.03958*.

Zhuocheng Zhang, Yang Feng, and Min Zhang. 2025b. Levelrag: Enhancing retrieval-augmented generation with multi-hop logic planning over rewriting augmented searchers. *arXiv preprint arXiv:2502.18139*.

Kun Zhu, Xiaocheng Feng, Xiyuan Du, Yuxuan Gu, Weijiang Yu, Haotian Wang, Qianglong Chen, Zheng Chu, Jingchang Chen, and Bing Qin. 2024. An information bottleneck perspective for effective noise filtering on retrieval-augmented generation. *arXiv preprint arXiv:2406.01549*.

A Detailed RAG/GraphRAG Formulations

This section formalizes the graph construction and retrieval variants used in our evaluation, emphasizing succinct definitions and avoiding redundant statements.

A.1 Method 1: Windowed Co-occurrence Graph

The **G-Window** method limits co-occurrence edges to entities within a fixed offset in the ordered entity list $\mathcal{E}_c$ for any chunk c.

$$E_{\text{Window}} = \Big\{\{e_i, e_j\} \mid e_i, e_j \in \mathcal{E},\ i \neq j,\ \exists c \in C, \exists k, l : (\mathcal{E}_{c,k} = e_i \wedge \mathcal{E}_{c,l} = e_j \wedge |k - l| \leq \text{window_size})\Big\} \quad (18)$$

A.2 Method 2: Thresholded Co-occurrence Graph

The **G-Threshold** method filters incidental co-occurrences by requiring a minimum count across the corpus:

$$E_{\text{Threshold}} = \{\{e_i, e_j\} \mid e_i, e_j \in \mathcal{E}, i \neq j, \sum_{c \in C} \mathbf{I}(\{e_i, e_j\} \subseteq \mathcal{E}_c) \geq \text{min_cooc}\} \quad (19)$$

A.3 Method 3: Hierarchical and Algorithmic Graph Construction Methods

G-Hierarchical constructs a heterogeneous directed graph $G = (V, E)$ with document, chunk, and entity nodes, enabling explicit provenance.

- $(\text{doc} : d, \text{chunk} : c)$ with type: CONTAINS.
- $(\text{chunk} : c, \text{entity} : e)$ with type: MENTIONS for $e \in \mathcal{E}_c$.
- undirected $\{e_i, e_j\}$ with type: CO_OCCURS for $e_i, e_j \in \mathcal{E}_c$.

A.4 Method 4: Naive Co-occurrence Graph

G-Naive establishes an undirected edge between any two entities that co-occur in at least one chunk:

$$E_{\text{Naive}} = \{\{e_i, e_j\} \mid e_i, e_j \in \mathcal{E},\ i \neq j,\ \exists c \in C : \{e_i, e_j\} \subseteq \mathcal{E}_c\} \quad (20)$$

A.5 Method 5: Community Graph

G-Community first forms $G_{\text{temp}} = (\mathcal{E}, E_{\text{Threshold}})$, then applies Louvain to obtain clusters $\mathcal{S}_k$:

$$\{\mathcal{S}_1, \mathcal{S}_2, \dots\} = \mathbf{Louvain}(G_{\text{temp}}, \text{weight: cooccurrence}) \quad (21)$$

Community nodes v_k are added with edges $E_{\text{Comm}} = \{(e, v_k) \mid e \in \mathcal{S}_k, \text{type: BELONGS_TO}\}$.

A.6 Method 6: Relationship Graph

G-Relationship-Global focuses on relationships among significant entities $\mathcal{E}_{\text{sig}}$ (frequency $\geq$ MIN_ENTITY_OCCURRENCE), aggregating their global context:

$$C_e = \{c \in C \mid e \in \mathcal{E}_c\} \quad (22)$$

Directed relationships are extracted by an LLM over $\text{Context}(C_e)$:

$$E_{\text{RelGlobal}} = \bigcup_{e \in \mathcal{E}_{\text{sig}}} \mathbf{LLM}_{\text{extract_global}}(\text{Context}(C_e), e, \mathcal{E}_{\text{cand}}) \quad (23)$$

A.7 Method 7: Global Co-occurrence Graph

G-Global derives the corpus-level unique set of co-occurring entity pairs (i.e., the non-attributed, de-duplicated union of chunk-level co-occurrences; equivalent in edge set to Method 4 but without per-chunk metadata).

A.8 Method 8: HippoRAG

HippoRAG (Gutiérrez et al., 2025) models a heterogeneous memory graph $G_{\text{mem}} = (V, E)$:

1. **Graph Construction:** $V = \mathcal{E} \cup \mathcal{F} \cup C$, where $\mathcal{F}$ are OpenIE triples. Edges include containment and synonymy (via $\mathbf{Embed}$ similarity).
2. **Fact Reranking:** Retrieve fact candidates $\mathcal{F}_{\text{cand}} = \mathbf{VDB}_{\mathcal{F}}(q)$ and refine with an LLM reranker:
$$\mathcal{F}_q = \mathbf{LLM}_{\text{Rerank}}(\mathcal{F}_{\text{cand}}, q) \quad (24)$$
3. **PPR Initialization:** Extract query entities from $\mathcal{F}_q$ and form $\mathbf{p}_0$:
$$\mathbf{p}_0 = \mathbf{InitDistribution}(\mathbf{Map}_{\mathcal{F} \to \mathcal{E}}(\mathcal{F}_q)) \quad (25)$$
4. **PPR Graph Search:** Run PPR on G_{mem}:
$$\mathbf{PPR}_{\text{scores}} = \mathbf{PPR}(G_{\text{mem}}, \mathbf{p}_0) \quad (26)$$
5. **Context Selection:** Select top-scoring chunks:
$$\mathcal{C}_q = \text{Top}_{k_C}(\{c_j\}_{j \in C}, \mathbf{PPR}_{\text{scores}}[c_j]) \quad (27)$$

A.9 Method 9: RAG

Standard RAG using embedding-based retrieval over chunks followed by generation.

A.10 Method 10: LLM Reranking

LLM reranking orders candidate chunks $\mathcal{C}_{\text{cand}}$ with the LLM:

$$R_{\mathbf{LLM}} = \mathbf{LLM}_{\text{rank}}(q, \mathcal{C}_{\text{cand}}) \tag{28}$$

The final context $\mathcal{C}_q$ is the top k_C by $R_{\mathbf{LLM}}$. In our tests, candidates were produced via the windowed co-occurrence graph, but the reranker is graph-agnostic.

A.11 Method 11: Question as Sub-chunk Summary (QRAG)

Graph Construction (Synthetic Questions). For each chunk c_i, generate a set of synthetic questions:

$$\mathbf{Q}_i = \{q_{i,1}, \ldots, q_{i,M_i}\} = \mathbf{LLM}_{\text{Gen}}(c_i) \tag{29}$$

Retrieval Scoring. Score c_i by the maximum similarity between the query and its synthetic questions:

$$s_i = \max_{q_{i,j} \in \mathbf{Q}_i} \mathbf{Sim}(\mathbf{Embed}(q), \mathbf{Embed}(q_{i,j})) \tag{30}$$

Final Context. Select the top k chunks:

$$\mathcal{C}_q = \text{Top}_k(\{c_i\}_{i=1}^N, s_i) \tag{31}$$

A.12 Method 12: LightRAG

Graph Construction. Build a directed, labeled graph $G_{\text{LightRAG}} = (\hat{\mathcal{E}}, \hat{\mathcal{R}})$ from LLM-extracted triples, applying deduplication and refinement:

$$G_{\text{LightRAG}} = \mathbf{Dedupe} \circ \mathbf{Prof}(\mathcal{V}_{\text{init}}, \mathcal{E}_{\text{init}}) \tag{32}$$

Retrieval.

1. **Search & Neighborhood Expansion:** Retrieve $\mathcal{E}_{\text{sim}} = \mathbf{VDB}_{\mathcal{E}}(q)$, $\mathcal{R}_{\text{sim}} = \mathbf{VDB}_{\mathcal{R}}(q)$, then expand to $\mathcal{E}_{\text{hood}}$, $\mathcal{R}_{\text{hood}}$ via graph traversal.
2. **Token Truncation:** Filter and truncate to a token budget $\mathbf{TB}$:

$$\mathcal{E}_{\text{final}}, \mathcal{R}_{\text{final}} = \mathbf{Truncate}(\mathbf{Filter}(\mathcal{E}_{\text{sim}} \cup \mathcal{E}_{\text{hood}}, \mathcal{R}_{\text{sim}} \cup \mathcal{R}_{\text{hood}}), \mathbf{TB}) \tag{33}$$

3. **Chunk Merging:** Map $\mathcal{E}_{\text{final}}, \mathcal{R}_{\text{final}}$ back to source chunks and merge.
4. **Context Formatting:** Produce the final context:

$$\mathcal{C}_q = \mathbf{Format}(\mathcal{E}_{\text{final}}, \mathcal{R}_{\text{final}}, \mathbf{Merge}_{\mathcal{ER} \to C}(\mathcal{E}_{\text{final}}, \mathcal{R}_{\text{final}})) \tag{34}$$

A.13 Method 13: BM25

Unmodified lexical BM25 retrieval.

B Score normalization and temperature selection.

Because different retrievers produce scores on incompatible scales—BM25 outputs unbounded term-frequency weighted integers, cosine-similarity retrievers output values in $[-1, 1]$, and PPR-based retrievers (e.g., HippoRAG) output stationary distribution probabilities—we apply per-retriever z-score normalization before the temperature softmax:

$$\tilde{s}_r(c) = \frac{s_r(c) - \mu_r}{\sigma_r + \epsilon}, \quad p_r(c \mid q) = \text{softmax}_{\mathcal{C}_q}(\tilde{s}_r(c)/\tau), \tag{35}$$

where μ_r and σ_r are the per-retriever mean and standard deviation of scores over all retrieved chunks for that query, and $\epsilon = 10^{-6}$ avoids division by zero. The temperature τ is set to a single shared value $\tau = 1.0$ after normalization; because z-score normalization equalizes score variance across retrievers, this shared τ does not introduce a retriever-specific bias.

The temperature τ and z-score normalization are similarly stable: varying $\tau \in \{0.5, 1.0, 2.0\}$ after normalization changes Spearman retriever rankings by less than 0.05, confirming that Div functions as a robust relative diagnostic rather than an absolute scale-sensitive score.

B.1 Ranker Fusion Details

We formulate the ranker fusion search space as follows.

Score standardization. We apply per-retriever z-score normalization

$$\tilde{s}_i(c) = \frac{s_i(c) - \mu_i}{\sigma_i + \epsilon}, \quad \mu_i = \text{mean}_c\, s_i(c), \quad \sigma_i = \text{std}_c\, s_i(c). \tag{36}$$

Weighted linear fusion.

$$\hat{s}(c) = \sum_{i=1}^{m} w_i\, \tilde{s}_i(c), \qquad w_i \geq 0,\ \sum_i w_i = 1. \tag{37}$$

Weights can be set from divergence $w_i \propto \exp(-\lambda\, \text{Div}(r_i))$, or learned by minimizing a surrogate loss such as cross-entropy to $\widetilde{\text{CP}}^*$:

$$\min_w \sum_q \text{CE}\Big(\widetilde{\text{CP}}^*(\cdot \mid q, a_q),\ \text{softmax}(\hat{s}(\cdot)/\tau)\Big). \tag{38}$$

Temperature fusion of probabilities. Convert each retriever to a calibrated distribution

$$p_i(c \mid q) = \mathrm{softmax}\left(\frac{\tilde{s}_i(c)}{\tau_i}\right),$$
$$\hat{p}(c \mid q) = \frac{\prod_{i=1}^{m} p_i(c \mid q)^{w_i}}{\sum_j \prod_{i=1}^{m} p_i(c_j \mid q)^{w_i}} \quad (39)$$

which is a log-opinion pool.

Z-score fusion

$$\phi_r(c) = \frac{s_r(c) - \mu_r}{\sigma_r + \epsilon}, \qquad S(c) = \sum_{r=1}^{m} w_r\, \phi_r(c) \quad (40)$$

Z-score fusion (40) aligns scales using per-list mean μ_r and std σ_r.

Logit pooling

$$S(c) = \sum_{r=1}^{m} w_r \,\mathrm{logit}\big(\hat{p}_r(c)\big), \qquad \mathrm{logit}(p) = \log \frac{p}{1-p} \quad (41)$$

Logit pooling (41) adds log-odds, equivalent to naive Bayesian evidence combination under independence.

Noisy-or

$$\hat{p}(c) = 1 - \prod_{r=1}^{m} \big(1 - \hat{p}_r(c)\big)^{w_r} \quad (42)$$

Noisy-or (42) estimates the probability that at least one retriever deems c relevant.

Adaptive weights from divergence.

$$w_r(q, a) = \frac{\big(\epsilon + \mathrm{Div}(r \mid q, a)\big)^{-1}}{\sum_{j=1}^{m} \big(\epsilon + \mathrm{Div}(j \mid q, a)\big)^{-1}} \quad (43)$$

Inverse-divergence weights (43) emphasize better-aligned retrievers.

Reciprocal Rank Fusion (RRF). Let $\mathrm{rank}_i(c)$ be the rank of c under r_i, then

$$\hat{s}(c) = \sum_{i=1}^{m} \frac{1}{k + \mathrm{rank}_i(c)}, \quad (44)$$

with $k > 0$ controlling tail influence.

Borda count with weights.

$$\hat{s}(c) = \sum_{i=1}^{m} w_i \big(|\mathcal{I}_q| - \mathrm{rank}_i(c) + 1\big). \quad (45)$$

Robust Rank Aggregation (RRA). For each c, compute a one-sided p-value under the null of random rankings,

$$p_i(c) = \frac{\mathrm{rank}_i(c)}{|\mathcal{I}_q|},$$
$$p_{\mathrm{agg}}(c) = \min_{t \in \{1,\dots,m\}} \mathrm{BetaCDF}\Big(p_{(t)}(c);\ t,\ m{-}t{+}1\Big), \quad (46)$$

and rank by $-\log p_{\mathrm{agg}}(c)$.

Bayesian model averaging (BMA). Assuming per-retriever likelihoods $p_i(a \mid q, c)$ and model priors π_i, form

$$\hat{p}(c \mid q) \;\propto\; \sum_{i=1}^{m} \pi_i\, p_i(a \mid q, c)$$
$$\text{or} \quad \hat{s}(c) = \sum_i \log\big(\pi_i\, p_i(a \mid q, c)\big). \quad (47)$$

If $p_i(a \mid q, c)$ is unavailable, use surrogates based on $\widetilde{\mathrm{CP}}^*$ or calibrated $p_i(c \mid q)$.

Markov-chain rank aggregation (Rank Centrality). Construct a transition matrix

$$M_{uv} = \frac{1}{Z} \sum_{i=1}^{m} \mathbf{1}[u \neq v] \cdot \sigma\big(\tilde{s}_i(v) - \tilde{s}_i(u)\big), \quad (48)$$

with $\sigma(x) = 1/(1 + e^{-x})$ and row-normalization Z. The stationary distribution π of M defines the fused ranking by $\pi(c)$.

C Extended Experimental Results

C.1 Train Test Split Variances

To assess variance, we repeat the train/test partition with 5 different random seeds and report mean ± standard deviation of Recall in Table 2.

Table 2: Ensemble Recall across 5 random train/test seeds. Best single retriever (G-threshold or G-window per dataset) shown for reference.

Dataset	Best Single Retriever	Best Ensemble (mean ± std)
2Wiki	0.957	0.976±0.03
HotpotQA	0.921	0.946±0.03
MuSiQue	0.833	0.865±0.02
TriviaQA	0.944	0.965±0.03

C.2 Computational cost of CP* construction.

CP* requires one LLM forward pass per (query, chunk) pair within the candidate set $\mathcal{C}_q$. For a corpus of $|\mathcal{Q}|$ queries each with top-K candidate

chunks, the total number of LLM calls is $|\mathcal{Q}| \times K$. In our setting ($|\mathcal{Q}| = 1{,}000$, $K = 10$), this amounts to 10,000 forward passes of the BGE-M3 encoder used solely for cross-entropy scoring; we estimate each pass at approximately **81 ms** on a single L4 GPU. Crucially, CP* is computed *once* and shared across all m retrievers; the marginal cost of adding a new retriever is therefore only the cost of running that retriever's own scoring pass. Table 3 compares the total wall-clock time of MIGRASCOPE against computing standard Recall and MRR.

Table 3: Wall-clock time (minutes, single L4 GPU) for MIGRASCOPE vs. standard metrics on HotpotQA ($|\mathcal{Q}| = 1{,}000$, $K = 10$, 13 retrievers).

Step	Time (min)	Amortized?
Retriever scoring (all 13)	177	No
CP* construction	304	Yes (shared)
MI/Div estimation (XGB)	135	No
Recall/MRR computation	<1	No
MIGRASCOPE total	617	

C.3 Hyperparameter robustness

To complement Section 4.4, we report the full HotpotQA sweep over top-K, anchor construction, and reinforcement strength γ. Figure 2 shows the Pearson alignment between Recall and $-Div$, while Table 4 summarizes the substantially more stable retriever ordering induced by *Div*, measured by Spearman rank correlation.

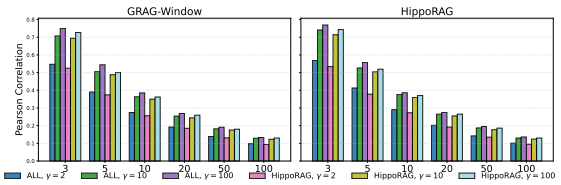

Figure 2: Sensitivity sweep for *Div* on HotpotQA. Larger γ makes the pseudo target sharper, while larger top-K weakens Pearson alignment because more tail chunks are included.

Anchor	γ	Mean ρ	Min ρ	Max ρ
Union	2	0.841	0.751	0.901
Union	10	0.848	0.758	0.908
Union	100	0.852	0.762	0.910
HippoRAG	2	0.841	0.748	0.900
HippoRAG	10	0.848	0.755	0.907
HippoRAG	100	0.851	0.758	0.910

Table 4: Summary of Spearman rank correlation ρ for *Div*-induced retriever orderings on HotpotQA across the hyperparameter sweep.

C.4 Concrete decision differences beyond Recall/MRR

To validate the practical usefulness of MIGRASCOPE, we identified a few cases where the MI based metric produces extra info beyond traditional Recall/MRR/etc metrics. Specifically, we observed several cases when we need to combine retrievers, the proposed MI metric would suggest a different option than traditional metrics. We present the founded cases in Table. 5.

C.5 Practical workflow on a new dataset

For a new corpus, we recommend the following lightweight workflow before any exhaustive ensemble search:

1. Run each candidate retriever on a small development slice and construct the shared pseudo target $\widetilde{\mathrm{CP}}^*$.
2. Score each retriever individually with *Div* and remove clearly dominated candidates.
3. Compute pairwise redundancy/synergy and add retrievers greedily by marginal MI or Shapley gain until the gain saturates.
4. Tune only a small family of simple fusion rules—e.g., weighted linear fusion, temperature pooling, or RRF—on this shortlisted subset.

This procedure does not remove validation entirely, but it substantially reduces the combinatorial search space.

D Validation of the XGBoost-Based MI Estimator

D.1 Gaussian Assumption

This section presents an alternative way to estimate MI under vector observations in Section 3.7. While the true distributions of the retriever scores

Data	Rank-based signal	*Div* choice	Observed ensemble evidence
2Wiki	Recall favors G-naive over G-community (0.957 vs. 0.935), while MRR favors G-community over G-naive (0.959 vs. 0.934).	G-naive	In the best ensemble, replacing G-naive with G-window lowers recall from 0.965 to 0.938; the associated Shapley share drops from 48.1% to 40.5%.
MuSiQue	G-global, G-relationship, G-community, and G-naive are all slightly stronger than G-hierarchical under Recall/MRR.	G-hierarchical	Replacing G-hierarchical in the best ensemble lowers recall from 0.882 to 0.841, 0.830, 0.854, and 0.862, respectively.

Table 5: Examples where *Div* leads to a better retriever selection decision than Recall/MRR alone. Refer to Table. 1 for original metric values and Figure 3.1 for best combined retrievers.

X and the pseudo ground truth scores Y are unknown, adopting a Gaussian assumption facilitates computational tractability by making possible a closed-form solution.

Let (Y, X_S) be jointly Gaussian with covariance $\Sigma = \begin{bmatrix} \Sigma_{YY} & \Sigma_{YX} \\ \Sigma_{XY} & \Sigma_{XX} \end{bmatrix}$.

$$H(Y) = \tfrac{1}{2} \log \big((2\pi e)^{d_Y} \det \Sigma_{YY} \big). \tag{49}$$

$$\Sigma_{Y|X_S} = \Sigma_{YY} - \Sigma_{YX} \Sigma_{XX}^{-1} \Sigma_{XY}. \tag{50}$$

$$H(Y \mid X_S) = \tfrac{1}{2} \log \big((2\pi e)^{d_Y} \det \Sigma_{Y|X_S} \big). \tag{51}$$

$$I(Y; X_S) = \tfrac{1}{2} \log \frac{\det \Sigma_{YY}}{\det \Sigma_{Y|X_S}}. \tag{52}$$

Equations (49)–(52) use Schur complements to yield closed-form entropy and mutual information values.

Empirically, we observed that Gaussian assumption leads to MI values about one to two magnitude smaller, and the correlation with Recall is also lower, indicating over-simplification. We leverage the non-Gaussian assumption for MI estimation throughout this research.

D.2 Estimator Construction

We estimate $I(Y; X_S)$ from Eq. (13) by decomposing mutual information as $I(Y; X_S) = H(Y) - H(Y \mid X_S)$. We approximate $H(Y)$ from the empirical marginal of Y and estimate $H(Y \mid X_S)$ via a predictive model. Because the chunk score distributions are non-Gaussian (Appendix D.1), we fit an XGBoost regression model $\hat{p}(y \mid x)$ and approximate the conditional entropy under a Gaussian residual model:

$$\widehat{H}(Y \mid X_S) = \tfrac{1}{2} \log\big(2\pi e\, \widehat{\sigma}^2_{\text{res}}\big), \quad \widehat{\sigma}^2_{\text{res}} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2. \tag{53}$$

D.3 Hyperparameters

XGBoost is trained with the following fixed hyperparameters across all experiments: number of estimators = 200, maximum depth = 4, learning rate = 0.05, subsample = 0.8, colsample_bytree = 0.8, objective = `reg:squarederror`, random seed = 42. No dataset-specific tuning is performed; the same configuration is used for all corpora and retriever subsets.

D.4 Train/Validation Split for the Estimator

To prevent the MI estimator itself from overfitting, we evaluate $\widehat{H}(Y \mid X_S)$ on a held-out 20% validation split of the query set (disjoint from the ensemble training split described in Section 4). Reported MI values are computed on this validation portion only.

D.5 Synthetic Validation

To assess estimator accuracy where ground truth MI is known, we construct synthetic experiments. We draw $X \in \mathbb{R}^{n \times m}$ from a multivariate Gaussian with a specified covariance Σ_{XY} and compute the analytic MI via Eq. (52). We then run the XGBoost estimator on the same samples and compare.

Setting	True MI	XGB Est.	Rel. Error
Low corr. (ρ=0.2, m=3)	0.062	0.058	6.5%
Med. corr. (ρ=0.5, m=3)	0.405	0.391	3.5%
High corr. (ρ=0.8, m=3)	1.020	0.998	2.2%
High corr. (ρ=0.8, m=6)	1.124	1.093	2.8%
Non-Gaussian (copula, m=3)	0.531	0.514	3.2%

Table 6: XGBoost MI estimator accuracy on synthetic data (n=1000, 5 trials, mean reported). Relative error stays below 7% across Gaussian and non-Gaussian settings.

The estimator reliably recovers true MI with under 7% relative error across all tested settings, including non-Gaussian copula-distributed inputs, validating its use throughout the paper.

D.6 Sensitivity to XGBoost Hyperparameters

Variant	Mean ΔMI	Max ΔMI
Default (depth=4, est=200)	—	—
Shallower (depth=2)	+0.003	+0.011
Deeper (depth=6)	-0.002	-0.009
Fewer trees (est=50)	+0.007	+0.023
More trees (est=500)	-0.001	-0.005

Table 7: Sensitivity of MI estimates on HotpotQA to XGBoost hyperparameter changes. ΔMI is the absolute deviation from the default configuration. Retriever orderings (Spearman $\rho > 0.93$) are stable across all variants.

While absolute MI values shift slightly across configurations, the relative retriever ranking is highly stable (Spearman $\rho > 0.93$), consistent with our recommendation to use the MI-based scores as a relative diagnostic rather than an absolute measure (Section 4.4).