

RCT-MARS: When Per-Query Retrieval Routing Fails, and What It Takes to Succeed

Ankit Srivastava

`ankits@ithena.ai`

ITHENA

Research Article

Keywords: Retrieval complexity, Algorithm selection, Query routing, Soft routing, Performance signatures, Multi-paradigm retrieval, Cross-encoder reranking

Posted Date: April 7th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9132181/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Journal of Intelligent Information Systems on August 6th, 2026. See the published version at <https://doi.org/10.1007/s10844-026-01080-5>.

RCT-MARS: When Per-Query Retrieval Routing Fails, and What It Takes to Succeed

Anonymous Author^{1*}

^{1*} Anonymous Institution, Anonymous, Anonymous.

Corresponding author(s). E-mail(s): anonymous@email.com;

Abstract

An oracle that selects the best retrieval paradigm per query substantially outperforms any fixed method, yet learned routers consistently fail to beat strong baselines—the *routing paradox*. We study this paradox across five retrieval paradigms (BM25, dense embeddings, knowledge-graph retrieval, agentic multi-step search, and cross-encoder reranking) and six benchmarks. **Hard routing** (discrete paradigm selection) fails severely: the best hard router (XGBoost-Direct, 0.387) falls 7.8 pp below the best fixed paradigm (Reranker, 0.465) despite an oracle ceiling of 0.599, closing -57.9% of the oracle gap. We then show **what it takes to succeed**: Corpus-Aware Soft Routing (CASR) replaces discrete selection with learned per-query fusion weights via XGBoost multi-output regression and temperature-scaled softmax ($\tau = 0.05$). CASR achieves 0.487 nDCG@10, significantly outperforming Dense (+4.9 pp, $p = 0.002$), numerically exceeding the best fixed paradigm (Reranker, +2.2 pp, $p = 0.112$), and matching unsupervised fusion (RRF, 0.482; $p = 0.755$)—closing +16.4% of the oracle gap versus -57.9% for hard routing—a 74 percentage-point swing (+16.4 – (−57.9)). To diagnose why hard routing fails and soft routing succeeds, **RCT-MARS** (Retrieval Complexity Taxonomy through Multi-paradigm Algorithm Routing and Selection) constructs performance-signature vectors, clusters them to discover three stable complexity classes (ARI = 0.773, 95% CI [0.529, 0.991]), and uses the taxonomy as a diagnostic lens. The key insight is that discrete paradigm selection requires precise information that query features alone cannot provide, whereas soft routing hedges across paradigms, tolerating prediction uncertainty while exploiting paradigm complementarity.

Keywords: Retrieval complexity, Algorithm selection, Query routing, Soft routing, Performance signatures, Multi-paradigm retrieval, Cross-encoder reranking

1 Introduction

Modern information retrieval now encompasses fundamentally different paradigms: sparse lexical matching (Robertson and Zaragoza, 2009), dense semantic retrieval (Karpukhin et al, 2020), knowledge-graph-augmented reasoning (Yasunaga et al, 2021), cross-encoder reranking (Nogueira and Cho, 2019), and agentic multi-step search (Yao et al, 2023; Trivedi et al, 2023), each grounded in different computational assumptions. This creates a *routing paradox*: empirical studies consistently show that different retrievers win on different queries, and an oracle that selects the best retriever per query can substantially outperform any fixed method. Yet per-query routing systems rarely improve over strong single baselines in practice.

This paradox raises three linked questions. First, *does retrieval complexity have structure across paradigms?* When we measure how the same query performs under BM25, dense embeddings, knowledge-graph retrieval, cross-encoder reranking, and agentic reasoning, do coherent complexity regimes emerge, i.e., groups of queries that share similar paradigm-response profiles? Second, if such regimes exist, *why can't learned routers exploit them?* And third, *what would it take for routing to succeed?* Despite the proliferation of routing systems, no prior work has systematically characterized the landscape of multi-paradigm retrieval complexity to answer these questions.

Prior work on understanding retrieval complexity is limited in three critical ways. First, **query performance prediction (QPP)** estimates a scalar difficulty score within a single system (Cronen-Townsend et al, 2002; Shtok et al, 2012; Carmel and Yom-Tov, 2010), offering no multi-paradigm perspective on complexity. Second, **per-query routing systems** operate within narrow paradigm spaces (typically choosing between sparse and dense retrieval (Arabzadeh et al, 2021)) or classify queries into pre-defined complexity tiers that conflate task complexity with retrieval method suitability (Jeong et al, 2024). These systems treat routing as an engineering objective, not as a lens for studying complexity structure. Third, the **algorithm selection** community (Rice, 1976) has developed sophisticated instance-based solver selection methods (e.g., SATzilla (Xu et al, 2008), ISAC (Kadioglu et al, 2010)), but these cluster on hand-crafted instance features rather than performance outcomes and have not been applied to information retrieval across paradigmatically diverse methods.

We present **RCT-MARS** (Retrieval Complexity Taxonomy through Multi-paradigm Algorithm Routing and Selection), an empirical study and methodological framework built on three key insights:

1. **Performance-signature vectors encode paradigm suitability.** For each query q , we construct a vector $\mathbf{s}(q) = [s_1(q), s_2(q), \dots, s_K(q)]$ where $s_k(q)$ is the retrieval effectiveness of paradigm k on query q . This extends QPP from scalar prediction to vector-valued paradigm profiling.
2. **Clustering performance signatures reveals emergent complexity classes.** Applying unsupervised clustering to the space of performance-signature vectors discovers groups of queries that share similar paradigm-response profiles. These clusters constitute *data-driven retrieval complexity classes*, defined not by intent or topic but by how the query interacts with fundamentally different retrieval mechanisms.

3. **Complexity classes expose the limits of learned routing.** Testing whether discovered classes can be exploited for per-query paradigm selection reveals a fundamental information deficit: query-level features carry insufficient information to predict paradigm suitability, even when the underlying complexity structure is real and stable.

The contributions of this paper are empirical and analytical rather than purely algorithmic; the individual pipeline components (performance vectors, k-means clustering, classifier training, multi-output regression) are standard; the novelty lies in diagnosing why hard routing fails, identifying why soft routing succeeds, and providing a principled framework for studying multi-paradigm retrieval complexity. Importantly, RCT-MARS characterizes routing over *practical retrieval configurations* under realistic indexing and compute constraints, not idealized paradigm upper bounds; the resulting complexity landscape reflects how retrieval systems actually perform in deployable settings:

1. **Diagnose the hard routing barrier.** Running five paradigms (BM25, Dense, KG, Agentic, cross-encoder Reranker) across six benchmarks ($N_{\text{test}} = 236$), the best hard router (XGBoost-Direct, 0.387) falls 7.8 pp below the best fixed paradigm (Reranker, 0.465) and is far worse than unsupervised fusion (RRF, 0.482), despite 29% oracle headroom (oracle: 0.599). Hard routing closes -57.9% of the oracle gap, meaning it *worsens* performance relative to the best individual paradigm.
2. **Show what it takes to succeed.** Corpus-Aware Soft Routing (CASR) replaces discrete paradigm selection with learned per-query fusion weights via XGBoost multi-output regression and temperature-scaled softmax. CASR achieves 0.487 nDCG@10, significantly outperforming Dense (+4.9 pp, $p = 0.002$), numerically exceeding the best fixed paradigm (Reranker, +2.2 pp, $p = 0.112$), and matching unsupervised fusion (RRF, 0.482; $p = 0.755$). CASR closes +16.4% of the oracle gap versus -57.9% for the best hard router (XGBoost-Direct)—a 74 percentage-point swing. We measure the swing against the best hard router because it represents the most favorable case for discrete paradigm selection; comparing against weaker routers would only widen the gap.
3. **Explain the gap as an information deficit.** The hard routing barrier reflects $I(P^*; C \mid Q)$: paradigm suitability depends on both query and corpus, but hard routers observe only the query. A four-level ablation ladder validates this interpretation: adding corpus-level features (especially retrieval probes) substantially improves routing, with L4 achieving +8.0% relative gain over query-only features. Soft routing overcomes the remaining deficit by hedging across paradigms rather than committing to one.
4. **Map the retrieval complexity landscape.** Performance-signature clustering across five paradigms reveals three stable complexity classes ($\text{ARI} = 0.773$, 95% CI [0.529, 0.991]) in a five-dimensional signature space. These clusters serve as a *diagnostic lens*, revealing paradigm dominance patterns and qualitatively interpretable query categories.

The remainder of this paper is organized as follows. Section 2 surveys related work across six research areas. Section 3 formalizes the RCT-MARS methodology. Section 4 details the experimental design. Section 5 presents results and analysis across

five research questions. Section 6 discusses implications, limitations, and future work. Section 7 concludes.

2 Related Work

RCT-MARS draws on and extends six distinct research areas. We survey each, identifying partial overlaps and articulating how our study differs.

2.1 Query Performance Prediction

The QPP literature spans two decades of predicting how well a single retrieval system handles a given query. Cronen-Townsend et al (2002) introduced the Clarity Score, measuring KL-divergence between query and collection language models as a pre-retrieval difficulty predictor. Shtok et al (2012) developed post-retrieval predictors using reference lists and the Unified Estimation Framework (UEF). Carmel and Yom-Tov (2010) codified the field in a comprehensive synthesis. All QPP work predicts a *unidimensional* easy/hard score within one system.

Two QPP-adjacent contributions come closer to our approach. Balasubramanian and Allan (2010) directly addressed “Learning to Select Rankers,” using QPP-style features to choose between learning-to-rank models on a per-query basis. Meng et al (2025) demonstrated ranker-agnostic QPP using LLM-generated relevance judgments, enabling cross-system performance prediction. Neither constructs a multi-paradigm performance vector or clusters queries by it.

He and Ounis (2004) noted that QPP predictors could support “query-based pre-retrieval model selection,” making them early conceptual ancestors of routing. However, this was never realized across fundamentally different paradigms.

Differentiation

QPP predicts one scalar per system; RCT-MARS constructs and clusters *multi-dimensional performance vectors* across paradigms.

2.2 Per-Query Retrieval Routing

Several systems dynamically select retrieval strategies, but all operate within constrained paradigm spaces. Arabzadeh et al (2021) trained a BERT classifier to route queries between sparse (BM25), dense (ANCE), or hybrid retrieval on MS MARCO, the closest existing IR routing work to RCT-MARS’s routing component. Adaptive-RAG (Jeong et al, 2024) classifies queries into three pre-defined complexity levels (no retrieval, single-step, multi-step) using a T5-Large classifier trained on which strategy produces correct answers. RouteRAG (Guo et al, 2025) uses GRPO-based reinforcement learning to route between text and graph retrieval at each reasoning step within a single generation pass.

More recent systems expand the routing space incrementally. SymRAG (Bhaskim et al, 2025) routes between neural, symbolic, and hybrid retrieval pathways using utility scores. LevelRAG (Li et al, 2025b) uses an LLM planner to decompose queries and

route sub-queries to sparse, dense, or web searchers. Bai et al (2025) built a rule-driven routing agent selecting among documents, databases, and direct LLM generation.

On the fusion side, Cormack et al (2009) established Reciprocal Rank Fusion (RRF). Mala et al (2025) introduced Dynamic Weighted RRF using query specificity to set per-query fusion weights. Hsu et al (2025) proposed Dynamic Alpha Tuning, where an LLM scores top-1 results from sparse and dense retrievers to set adaptive weights. MFAR (Li et al, 2025a) learns query-conditioned weighting across dense and lexical indexes.

Differentiation

Existing routing systems use two to three paradigms without discovering emergent query categories. RCT-MARS spans five fundamentally different paradigms and adds clustering to produce interpretable complexity classes, then tests whether these classes can be exploited via hard or soft routing.

2.3 Federated Search and Resource Selection

Federated search provides the conceptual ancestor for routing queries to different “resources,” but the resources are always data collections, never retrieval methods. Callan (2000) formalized distributed IR with the CORI algorithm. Si and Callan (2003) proposed ReDDE for estimating relevant document distributions across databases. Shokouhi and Si (2011) surveyed the field comprehensively. More recently, FeB4RAG (Wang et al, 2024) evaluated federated search in RAG contexts, and RAGRoute (Guerraoui et al, 2025) built a lightweight neural classifier for dynamic data source selection.

Differentiation

Federated search selects among data collections using one retrieval method; RCT-MARS selects among retrieval methods on the same data.

2.4 Algorithm Selection

The algorithm selection literature provides the strongest structural analogs to RCT-MARS. Rice (1976) formalized the Algorithm Selection Problem (ASP): given a problem instance characterized by features, select the best algorithm from a portfolio. SATzilla (Xu et al, 2008) operationalized this for SAT solvers using per-solver runtime regression models. AutoFolio (Lindauer et al, 2015) automated the configuration of algorithm selectors.

ISAC (Kadioglu et al, 2010) is the single most structurally similar prior work. ISAC clusters training instances into homogeneous subsets using g -means on feature vectors, configures a solver for each cluster, then assigns new instances to the nearest cluster. This is architecturally identical to RCT-MARS’s “cluster queries, assign paradigm per cluster, route new queries.” The **critical difference**: ISAC clusters on *hand-crafted instance features*, while RCT-MARS clusters on *performance-signature vectors* (the vector of per-paradigm effectiveness scores). This is not a minor distinction: clustering by performance signatures means the clusters directly encode which paradigms work

well together and which do not, rather than relying on a feature extraction step that may miss relevant dimensions.

Smith-Miles’ Instance Space Analysis (Smith-Miles et al, 2023) visualizes algorithm performance landscapes by projecting instances into 2D feature spaces, but has not been applied to IR paradigm selection.

Within IR, Beel and Kothhoff (2019) organized the first AMIR workshop on Algorithm Selection and Meta-Learning in Information Retrieval, explicitly noting that “the IR community has paid little attention to the algorithm selection problem.” Arora and Yates (2019) presented a meta-learner using axiomatic IR features to select between pairs of retrieval methods, the closest IR-specific algorithm selection work, but limited to pairwise selection without clustering. Deveaud et al (2016, 2018) cast per-query system configuration selection as learning-to-rank over configurations varying retrieval model parameters. Mothe and Ullah (2023) extended this with risk-sensitive selection, referencing He and Ounis (2004), who proposed clustering queries and associating a configuration with each cluster, the earliest IR instance of the cluster-then-route pattern, though within a single paradigm family.

Differentiation

Unlike ISAC, RCT-MARS clusters on performance signatures rather than instance features. Unlike Arora and Yates (2019), it spans paradigmatically diverse methods and adds emergent-class discovery.

2.5 Query Taxonomies and Complexity Characterization

Existing query classification schemes are manually defined. Broder (2002) proposed the navigational / informational / transactional taxonomy. Rose and Levinson (2004) refined it with subcategories. Jansen et al (2008) operationalized automatic classification. These taxonomies capture user *intent*, not retrieval *complexity*. A query can be informational yet trivially retrievable, or navigational yet requiring multi-hop reasoning.

BRIGHT (Su et al, 2025) introduced the first retrieval benchmark requiring intensive reasoning, demonstrating that state-of-the-art dense retrievers scoring 59.0 nDCG@10 on MTEB drop to 18.3 on reasoning-intensive queries. BRIGHT implicitly recognizes varying retrieval complexity but does not formalize complexity classes. RouterRetriever (Lee et al, 2025) routes queries to domain-specific expert LoRA modules, but routes by domain, not complexity. MoR (Liu et al, 2025) provides the strongest empirical validation of RCT-MARS’s premise: with an oracle selecting the best retriever per query, a mixture of small retrievers outperforms individual 7B models by +3.9%, confirming that different retrievers win on different query subsets. LTRR (Chen et al, 2025) uses pairwise XGBoost to rank retrieval systems per query based on downstream generation quality.

Differentiation

Existing taxonomies define categories *a priori* by intent or complexity level; RCT-MARS discovers them *a posteriori* from performance data.

2.6 Self-Discovery and RL-Based Routing

No retrieval system discovers its own operating modes through performance-profile clustering. CAME (Fan et al, 2025) uses competitive learning to specialize MoE retrieval experts, but all experts are dense retrievers, not different paradigms. DSPy (Khattab et al, 2024) compiles declarative LM pipelines with composable retrieval modules, but optimizes prompts and demonstrations, not paradigm selection.

On the RL side, Search-R1 (Jin et al, 2025) and R1-Searcher (Song et al, 2025) train LLMs via RL to formulate search queries but invoke only a single search engine. Router-R1 (Zhang et al, 2025) uses RL to learn multi-round routing policies among LLMs, the closest RL routing work, though routing targets are generation models, not retrieval paradigms. R1-Router (Wang et al, 2025) learns to route across knowledge bases during multimodal reasoning. Glowacka (2019) surveyed bandits for document ranking in IR, but existing bandit work focuses on document-level decisions, not paradigm-level routing.

A complementary line of work resolves retrieval decisions at generation time rather than pre-retrieval. Self-RAG (Asai et al, 2024) trains an LM to decide *when* to retrieve and how to critique retrieved passages during generation, while CRAG (Yan et al, 2024) uses a lightweight evaluator to assess retrieval quality and triggers corrective actions (web search fallback, query rewriting) based on generation confidence. Both systems defer the routing decision until partial results are available, fundamentally different from RCT-MARS’s one-shot pre-retrieval paradigm selection. Their success suggests that interleaving retrieval with generation may partially resolve the information deficit we identify, since generation-time routing observes document content that pre-retrieval routers cannot.

Differentiation

RL and bandit approaches provide routing *mechanisms*, but none has been applied across fundamentally different retrieval paradigms. Self-RAG and CRAG resolve routing at generation time rather than pre-retrieval. RCT-MARS’s contribution is the empirical characterization of complexity regimes; any routing method, including generation-time approaches, could implement the dispatch layer.

2.7 Summary of Novelty

Table 1 summarizes the relationship between RCT-MARS and the most closely related prior work. No prior contribution combines paradigm breadth (≥ 5 methods), performance-signature clustering, emergent complexity classes, empirical diagnosis of the hard routing barrier, and a soft routing solution (CASR).

3 Methodology

We formalize the RCT-MARS methodology as an instantiation of the Algorithm Selection Problem (Rice, 1976) adapted for multi-paradigm information retrieval.

Table 1 Comparison of RCT-MARS with the most closely related prior work. Columns indicate whether each work includes multi-paradigm profiling (≥ 4 paradigms), performance-signature clustering, and emergent complexity class discovery.

Work	Overlap with RCT-MARS	Multi-paradigm	Perf.-sig. clustering	Emergent classes
ISAC (Kadioglu et al, 2010)	Cluster-then-route architecture	×	×	×
Arabzadeh et al. (2021)	Per-query routing	×	×	×
Adaptive-RAG (Jeong et al, 2024)	Complexity-based routing	×	×	×
Arora & Yates (2019)	Meta-learning for IR methods	×	×	×
Deveaud et al. (2016)	Per-query system selection	×	×	×
MoR (2025)	Per-query paradigm advantage	×	×	×
SATzilla (Xu et al, 2008)	Per-instance algorithm selection	×	×	×
Smith-Miles ISA (2023)	Instance space visualization	×	×	×
RCT-MARS (Ours)	All three + routing diagnosis	✓	✓	✓

The methodology comprises four stages: multi-paradigm evaluation, performance-signature construction, complexity class discovery, and learned routing (tested as an exploitability probe).

3.1 Problem Formulation

Let $\mathcal{Q}$ be a query space and $\mathcal{P} = \{P_1, P_2, \dots, P_K\}$ be a set of K retrieval paradigms, where each P_k implements a fundamentally different retrieval mechanism. In our instantiation ($K = 5$):

- P_1 : Sparse lexical retrieval (BM25)
- P_2 : Dense semantic retrieval (E5-Large-v2)
- P_3 : Knowledge-graph-augmented retrieval (Wikidata5M subgraph scoring)
- P_4 : Agentic multi-step retrieval (ReAct with LLM)
- P_5 : Cross-encoder reranking (BGE-Reranker-v2-M3 over Dense top-100)

For each query $q \in \mathcal{Q}$, corpus C , and paradigm P_k , let $m(q, P_k, C) \in \mathbb{R}$ denote a retrieval effectiveness metric (e.g., nDCG@10). We write $m(q, P_k)$ when the corpus is fixed or clear from context. The **Algorithm Selection Problem for retrieval** is:

$$P^*(q, C) = \arg \max_{P_k \in \mathcal{P}} m(q, P_k, C) \quad (1)$$

Solving Equation (1) exactly requires evaluating all paradigms on every query, which is computationally prohibitive at inference time but can be done offline to characterize complexity structure. The explicit dependence on C is critical: the same query may favor different paradigms on different corpora. A learned router $R(q)$ that observes only the query faces a fundamental *information deficit*: the gap between the information available to the router and the information needed to solve Equation (1). We formalize this in Section 6.2. RCT-MARS exploits the offline evaluation through a three-stage methodology: signature construction, clustering, and routing.

3.2 Performance-Signature Construction

Definition 1 (Performance-Signature Vector) For a query q , the performance-signature vector is:

$$\mathbf{s}(q) = [m(q, P_1), m(q, P_2), \dots, m(q, P_K)] \in \mathbb{R}^K \quad (2)$$

Each component $m(q, P_k)$ is computed by running paradigm P_k on query q against the target corpus and evaluating against ground-truth relevance judgments. To ensure comparability across paradigms with different score scales, we apply per-paradigm z -score normalization:

$$\hat{s}_k(q) = \frac{m(q, P_k) - \mu_k}{\sigma_k}, \quad \text{where } \mu_k = \mathbb{E}_q[m(q, P_k)], \sigma_k = \text{Std}_q[m(q, P_k)] \quad (3)$$

The normalized performance-signature matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{Q}| \times K}$ serves as the input to the clustering stage. The low dimensionality ($K = 5$) is by construction: each dimension represents a fundamentally different retrieval mechanism, enabling fully interpretable cluster centroids. This is the key departure from ISAC (Kadioglu et al, 2010), which clusters on hand-crafted instance features.

3.3 Emergent Complexity Class Discovery

We apply unsupervised clustering to the normalized signature matrix $\mathbf{S}$ to discover groups of queries with similar paradigm-response profiles.

Clustering Algorithm

We adopt a two-stage approach:

1. **Consensus cluster count determination.** We evaluate five independent criteria across $k \in [2, 14]$: the gap statistic (Tibshirani et al, 2001), silhouette analysis (Rousseeuw, 1987), Bayesian Information Criterion via Gaussian Mixture Models,

the elbow method (inertia), and the Calinski-Harabasz index. Each criterion votes for its optimal k . When no majority exists, we use the median of the votes as a tiebreaker—a robust choice that avoids both the extremes of individual criteria and the instability of mode-based selection.

2. **Final clustering.** Given the determined k^* , we apply k -means++ (Arthur and Vassilvitskii, 2007) with multiple random restarts, selecting the partition with the lowest within-cluster sum of squares.

Definition 2 (Retrieval Complexity Class) A retrieval complexity class $\mathcal{C}_c$ ($c = 1, \dots, k^*$) is a cluster of queries whose performance-signature vectors are mutually proximate:

$$\mathcal{C}_c = \{q \in \mathcal{Q} \mid \text{cluster}(q) = c\} \quad (4)$$

Each class is characterized by its centroid signature $\mu_c = \frac{1}{|\mathcal{C}_c|} \sum_{q \in \mathcal{C}_c} \hat{\mathbf{s}}(q)$ and its dominant paradigm $P_c^* = \arg \max_k \mu_{c,k}$.

Stability Analysis

We assess cluster stability through bootstrap resampling (sampling 80% of queries 1000 times), measuring adjusted Rand index (ARI) between bootstrap clusterings and the full-data partition. Stable complexity classes should achieve $\text{ARI} > 0.7$.

3.4 Learned Routing

Given discovered complexity classes, we train a lightweight router $R : \mathcal{Q} \rightarrow \{1, \dots, C\}$ that maps unseen queries to their predicted complexity class (and thereby to the class’s dominant paradigm). Critically, the router operates on *query features* $\mathbf{f}(q)$, not performance signatures, since computing performance signatures for a new query would require running all paradigms, defeating the purpose of routing.

Feature Representation

We extract the following query features:

- **Lexical:** query length, type-token ratio, entity count, numeric token count
- **Syntactic:** dependency parse depth, clause count, question-word type
- **Semantic:** query embedding (from a frozen encoder), embedding norm, cosine similarity to corpus centroid
- **QPP-inspired:** Clarity Score (Cronen-Townsend et al, 2002), query scope (IDF statistics), pre-retrieval difficulty predictors
- **Structural:** presence of comparison operators, aggregation keywords (“average,” “total”), temporal references, relational phrases

Router Architectures

We evaluate three routing strategies with distinct inductive biases:

1. **XGBoost classifier:** A gradient-boosted tree trained on tabular query features $\mathbf{f}(q)$ to predict cluster labels. Effective for capturing non-linear interactions between lexical and structural features.

2. **BERT classifier:** A fine-tuned BERT model that classifies queries directly from text into complexity classes. Captures distributional semantics not available in hand-crafted features.
3. **Nearest-Centroid (NC) router:** An MLP that maps query embeddings $\mathbf{f}(q)$ into the K -dimensional signature space via a learned projection $g : \mathbb{R}^d \rightarrow \mathbb{R}^K$, then assigns to the nearest cluster centroid $\boldsymbol{\mu}_c$. Uniquely, this architecture also supports *direct paradigm prediction* by taking $\arg \max$ of the predicted signature vector, bypassing centroids entirely.

Training Objectives

XGBoost and BERT are trained with cross-entropy loss on cluster labels. The NC router is trained with mean-squared error to predict performance-signature vectors:

$$\mathcal{L}_{\text{NC}} = \sum_{q \in \mathcal{Q}_{\text{train}}} \|g(\mathbf{f}(q)) - \hat{\mathbf{s}}(q)\|^2 \quad (5)$$

where $\hat{\mathbf{s}}(q)$ is the z-normalized performance signature. At inference, the predicted signature is compared to cluster centroids for class assignment.

3.5 Corpus-Aware Soft Routing (CASR)

The hard routing framework above selects a single paradigm per query ($\arg \max$), which demands precise information about paradigm suitability. We propose **Corpus-Aware Soft Routing (CASR)**, which replaces discrete selection with learned per-query fusion weights.

Multi-output regression

Instead of predicting a class label, CASR trains an XGBoost multi-output regressor to predict the full performance-signature vector $\hat{\mathbf{s}}(q) \in \mathbb{R}^K$ from corpus-aware query features $\mathbf{f}(q)$. The feature set includes all query features from Section 3.4 augmented with corpus identity (one-hot), corpus statistics (log corpus size, average document length), and retrieval probes (BM25/Dense top-1 scores).

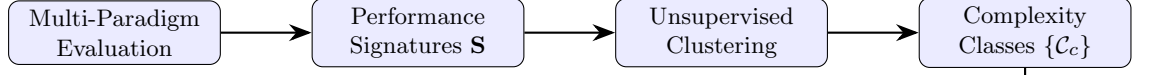
Temperature-scaled softmax

The predicted signature is converted to fusion weights via temperature-scaled softmax:

$$w_k(q) = \frac{\exp(\hat{s}_k(q)/\tau)}{\sum_{j=1}^K \exp(\hat{s}_j(q)/\tau)}, \quad k = 1, \dots, K \quad (6)$$

where $\tau > 0$ is a temperature parameter. Low τ concentrates weight on the predicted-best paradigm (approaching hard routing), while high τ distributes weight uniformly (approaching equal-weight fusion). We select τ by grid search on the router-training set.

Training (Offline)



Inference (Online)

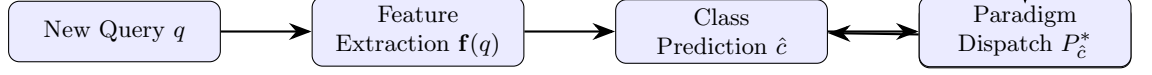


Fig. 1 The RCT-MARS methodology. **Top:** Offline phase constructs performance signatures across all paradigms, clusters them to discover complexity classes, and trains a routing classifier. **Bottom:** Online inference extracts query features, predicts the complexity class, and dispatches to the associated paradigm.

Weighted score fusion

For each query, the final ranked list is produced by fusing paradigm scores:

$$\text{score}_{\text{CASR}}(q, d) = \sum_{k=1}^K w_k(q) \cdot \tilde{s}_k(q, d) \quad (7)$$

where $\tilde{s}_k(q, d)$ is the min-max normalized retrieval score from paradigm P_k for document d .

Why soft routing succeeds

CASR tolerates prediction uncertainty in two ways: (1) the softmax hedges across paradigms rather than committing to one, so a slightly wrong prediction still assigns substantial weight to the true best paradigm; (2) multi-output regression predicts the *relative* paradigm performance profile, which is an easier learning target than identifying the single best paradigm.

3.6 Complete RCT-MARS Pipeline

Figure 1 illustrates the full methodology. The analysis phase (offline) constructs performance signatures, discovers complexity classes, trains hard routing classifiers, and fits the CASR soft router. The prediction phase (online) extracts features from an unseen query and either predicts its complexity class for hard routing or computes per-paradigm fusion weights for soft routing.

4 Experimental Framework

4.1 Research Questions

We organize our evaluation around five research questions:

RQ1 (Complementarity): Do retrieval paradigms exhibit complementary performance patterns across queries, and can these patterns be captured in structured performance signatures?

RQ2 (Emergence): Do emergent complexity classes arise from unsupervised clustering of performance signatures, and are they stable and interpretable?

RQ3 (Hard Routing): To what extent can hard routing (discrete paradigm selection) exploit the discovered complexity structure? What are the barriers to closing the oracle gap?

RQ4 (Taxonomy Role): What is the operational role of the cluster taxonomy—does it improve routing, or is its value purely diagnostic?

RQ5 (Soft Routing): Can corpus-aware soft routing (CASR) overcome the hard routing barrier identified in RQ3?

4.2 Retrieval Paradigm Implementations

We instantiate five core retrieval paradigms spanning distinct retrieval strategies, plus a hybrid fusion baseline and supplementary ColBERT late-interaction retrieval:

1. **Sparse Lexical (BM25):** Pyserini-based BM25 with parameters $k_1 = 0.9$, $b = 0.4$, representing the classical term-matching paradigm.
2. **Dense Semantic:** E5-Large-v2 (Wang et al, 2022) bi-encoder (1024-dim) with FAISS flat index, representing learned semantic similarity. The top-100 retrieved document overlap with BM25 is only 16.8% (Jaccard), confirming these are genuinely complementary retrieval mechanisms.
3. **Knowledge Graph (KG):** A subgraph-based retrieval pipeline built on the Wikidata5M knowledge graph (Wang et al, 2021). Given a query, spaCy NER extracts entity mentions, which are linked to Wikidata5M entities via exact label matching. A breadth-first traversal (up to 3 hops) expands the entity set, and documents are scored by their structural proximity to query entities: $\text{score}(d) = \sum_{e \in \text{entities}(d)} \frac{1}{1 + \text{hop}(e)}$, where $\text{hop}(e)$ is the shortest-path distance from a query entity to entity e in the KG. Document-entity associations are pre-computed via corpus-specific strategies: title-to-entity mapping for Wikipedia passages, Aho-Corasick multi-pattern matching for MS MARCO, and spaCy NER with entity linking for BEIR corpora. This paradigm uses *no* BM25 component; scoring is purely graph-structural, making it genuinely independent from the lexical retrieval paradigm.
4. **Agentic Multi-Step:** A ReAct-style agent (Yao et al, 2023) powered by Qwen2.5-Coder 32B via Ollama (local inference), with access to BM25 and Dense as retrieval tools, allowed up to 5 iterative search-reason cycles. The agent decomposes complex queries into sub-queries and aggregates retrieved passages. This paradigm wraps the underlying retrieval tools but produces substantially different ranked lists (Jaccard overlap with Dense: 18.0%).
5. **Cross-Encoder Reranker:** BGE-Reranker-v2-M3 (Chen et al, 2024), a 560M-parameter cross-encoder (24-layer XLM-RoBERTa) that reranks the Dense top-100 candidates via full query-document cross-attention. Unlike bi-encoder Dense retrieval, which independently encodes queries and documents, the cross-encoder jointly attends to query-document pairs, capturing fine-grained relevance signals. The Reranker achieves 0.465 nDCG@10, the highest among all individual

paradigms, demonstrating that cross-attention over Dense candidates substantially improves ranking quality (Dense: 0.438, +2.7 pp).

6. **Hybrid Fusion:** A score-level interpolation of BM25 and Dense retrieval scores with equal weight ($\alpha = 0.5$): $\text{score}_{\text{hybrid}}(q, d) = \alpha \cdot \text{score}_{\text{BM25}}(q, d) + (1 - \alpha) \cdot \text{score}_{\text{Dense}}(q, d)$. This non-learned fusion baseline tests whether simple score combination can outperform individual paradigms and learned routing.
7. **ColBERT Late-Interaction:** ColBERTv2 (Santhanam et al, 2022) with per-token late-interaction scoring, evaluated on BEIR corpora (FiQA, NFCorpus) only due to indexing constraints on large corpora. ColBERT achieves 0.416 nDCG@10 on the BEIR subset.

Behavioral independence

The Agentic paradigm uses BM25 and Dense as sub-tools, but produces substantially different ranked lists (18.0% Jaccard overlap with Dense). The Reranker reranks Dense top-100, so its candidate set is a subset of Dense’s; however, cross-attention reranking produces substantially different score distributions and ranking orders, constituting a paradigmatically different retrieval mechanism (bi-encoder vs. cross-encoder). The KG paradigm is *fully independent*: scoring is purely graph-structural with no lexical component. Although KG’s overall nDCG@10 is modest (0.239), this is expected for a highly specialized paradigm: KG performs well on entity-rich multi-hop queries (e.g., HotpotQA: 0.377) while contributing near-zero on lexically dominated corpora. Its inclusion is essential precisely because it reveals a distinct complexity class ($\mathcal{C}_1$, 23.3% under balanced coverage) invisible to BM25/Dense alone. We acknowledge the Agentic and Reranker overlaps as a limitation (Section 6.3).

Dense index construction

On large corpora (NQ, HotpotQA, MS MARCO), Dense indices are constructed from BM25-pooled candidate sets rather than exhaustive encoding of the full corpus. This means Dense recall is bounded by BM25 recall, and the Reranker (which reranks Dense top-100) inherits this ceiling. Consequently, the Reranker’s 0.465 overall score may understate its true ceiling on these corpora, and the oracle gap denominator is correspondingly conservative. On BEIR corpora (FiQA, NFCorpus), full corpus encoding is feasible, so this constraint does not apply. We note this as a practical limitation that affects the absolute performance levels but not the relative comparisons central to our analysis.

The five core paradigms (BM25, Dense, KG, Agentic, Reranker) are used for performance-signature construction and clustering. Hybrid fusion and ColBERT serve as additional baselines in the routing comparison.

Dense index construction

All dense indices use E5-Large-v2 (Wang et al, 2022) (1024-dim) as the encoding model. To make full-corpus dense indexing tractable, we build per-benchmark FAISS indices from BM25 passage pools rather than encoding the entire corpus. On smaller corpora (FiQA: 57K documents, NFCorpus: 3.6K documents), this pool covers the full

corpus and Dense performs well (FiQA: 0.536, NFCorpus: 0.475). On larger corpora the BM25 pool provides partial coverage, but Dense still achieves competitive scores (NQ: 0.501, HotpotQA: 0.432, MS MARCO: 0.379) due to the high quality of the E5-Large-v2 embeddings. Dense achieves 0.438 nDCG@10 overall; the cross-encoder Reranker applied to Dense top-100 further improves this to 0.465, making it the best individual paradigm.

Paradigm scope

We deliberately exclude learned sparse retrieval (e.g., SPLADE (Formal et al, 2021)) because it occupies the same lexical-semantic continuum as BM25 and Dense: SPLADE’s learned term expansion bridges the gap between exact matching and semantic similarity, meaning its inclusion would add a sixth paradigm without adding a *paradigmatically distinct* retrieval mechanism. Our five paradigms span fundamentally different retrieval strategies (lexical, semantic, structural, agentic, cross-encoder), not all possible retrieval models. Adding SPLADE as a sixth paradigm is a natural extension for future work.

4.3 Benchmarks and Datasets

We design a heterogeneous benchmark suite spanning six datasets to stress-test the discovery of complexity classes across diverse information needs (Table 2):

Table 2 Benchmark datasets for RCT-MARS evaluation. Queries are split 60/20/20 into Signature, Router-Train, and Test sets. Relevance judgments are constructed via paradigm-union pooling (NQ, HotpotQA, MS MARCO), official NIST judgments (TREC-DL), or BEIR relevance annotations (FiQA, NFCorpus).

Dataset	Domain	Total	Test	Avg Rels/q	Expected diversity
Natural Questions	Open-domain	3,610	567	40.4	BM25/Dense competitive
HotpotQA	Multi-hop	7,405	1,165	2.0	Dense/KG competitive
MS MARCO (dev)	Web search	10,047	1,582	1.1	Broad variation
TREC-DL 2019	Web search	43	8	95.4	BM25 advantage
FiQA	Financial QA	648	101	2.6	Domain-specific
NFCorpus	Medical	323	52	38.2	Specialized retrieval
Total		22,076	3,478		

TREC-DL 2019 (Craswell et al, 2020) uses the MS MARCO passage corpus with NIST-assessed relevance judgments, providing the only benchmark with professionally curated graded relevance. Its test split (8 queries after balanced filtering) is too small for statistical analysis; we include it only for paradigm-profile characterization and do not draw statistical conclusions from it.¹ FiQA (648 queries, 57K documents) and NFCorpus (323 queries, 3.6K documents) are drawn from the BEIR benchmark suite

¹All statistical tests and routing evaluations exclude TREC-DL due to insufficient test-set size. Its inclusion in the benchmark suite serves to illustrate BM25’s dominance on term-rich corpora with thorough relevance pooling.

(Thakur et al, 2021), adding domain-specific financial and medical retrieval challenges. These BEIR corpora also serve as the evaluation substrate for the supplementary ColBERT paradigm.

Data Splitting

To prevent leakage, we adopt a strict three-way split: 60% Signature Set (for computing performance signatures and z-normalization statistics), 20% Router-Train Set (for training routers on cluster labels), and 20% Test Set (for final evaluation). Z-normalization mean and standard deviation are computed exclusively on the Signature Set.

4.4 Baselines

We compare RCT-MARS against the following baselines, spanning fixed-paradigm, fusion, adaptive, and oracle approaches:

1. **Fixed-paradigm baselines:** Each individual paradigm applied uniformly to all queries (5 baselines).
2. **Hybrid fusion:** BM25+Dense score interpolation ($\alpha = 0.5$), testing whether simple unsupervised fusion outperforms learned routing.
3. **Reciprocal Rank Fusion (RRF):** (Cormack et al, 2009) combining all paradigms’ ranked lists with equal weights ($k = 60$).
4. **Arabzadeh et al. routing:** BERT-based sparse/dense/hybrid router (Arabzadeh et al, 2021), retrained on our paradigm set.
5. **Adaptive-RAG:** BERT classifier with three complexity levels (simple, moderate, complex), each mapped to a paradigm (Jeong et al, 2024).
6. **LLM Router:** Qwen2.5-Coder 32B prompted to select the best paradigm directly from the query text.
7. **ColBERT:** ColBERTv2 (Santhanam et al, 2022) late-interaction retrieval, evaluated on BEIR corpora only (see Section 4.2).
8. **CASR (Corpus-Aware Soft Routing):** Our proposed soft routing method (Section 3.5), producing per-query fusion weights via multi-output regression and temperature-scaled softmax.
9. **Oracle:** Per-query best paradigm selection (upper bound).

4.5 Evaluation Metrics

We evaluate at three levels:

- **Retrieval quality:** nDCG@{5, 10, 20}, Recall@{100, 1000}, MAP
- **Routing accuracy:** Class prediction accuracy, paradigm selection accuracy, oracle gap (percentage of oracle performance achieved)
- **Clustering quality:** Silhouette coefficient, adjusted Rand index (ARI) vs. bootstrap, Calinski-Harabasz index, and qualitative cluster interpretability
- **Statistical significance:** Permutation tests (10,000 permutations) on per-query nDCG@10 score differences for pairwise system comparisons, with 95% bootstrap

confidence intervals (10,000 resamples). Cluster stability assessed via bootstrap resampling (1,000 iterations, 80% subsampling) with 95% confidence intervals on ARI.

5 Results and Analysis

We present results organized by research question, followed by ablation analysis.

5.1 RQ1: Paradigm Complementarity

Table 3 shows per-paradigm nDCG@10 across the six benchmarks. The cross-encoder Reranker achieves the highest overall score (0.465), followed closely by Dense (0.438). The Reranker leads on four benchmarks (HotpotQA: 0.454, MS MARCO: 0.431, FiQA: 0.629, TREC-DL: 0.469), while Dense leads on NQ (0.501) and NFCorpus (0.475), and BM25 dominates TREC-DL with thorough relevance pooling (0.934). Despite the Reranker’s overall strength, paradigm complementarity remains: KG performs strongly on multi-hop queries (HotpotQA: 0.377), Agentic provides competitive performance on entity-rich benchmarks, and different paradigms are oracle-optimal on substantial query subsets. Supplementary ColBERT late-interaction retrieval achieves 0.416 on the BEIR subset.

Table 3 Per-paradigm nDCG@10 across benchmarks (balanced 5-paradigm set). Best single-paradigm score per benchmark in **bold**. The cross-encoder Reranker achieves the highest overall score by reranking Dense top-100 via full cross-attention. ColBERT is evaluated only on BEIR corpora (FiQA, NFCorpus).

Paradigm	NQ	HotpotQA	MSMARCO	TREC-DL	FiQA	NFCorpus	Overall
BM25	0.163	0.413	0.270	0.934	0.223	0.349	0.349
Dense	0.501	0.432	0.379	0.220	0.536	0.475	0.438
KG	0.005	0.377	0.007	0.000	0.021	0.190	0.239
Agentic	0.216	0.373	0.078	0.000	0.173	0.311	0.285
Reranker	0.476	0.454	0.431	0.469	0.629	0.381	0.465
Hybrid	0.449	0.497	0.392	0.612	0.517	0.421	0.474
ColBERT	—	—	—	—	0.434	0.365	0.416 [†]
Oracle	—	—	—	—	—	—	0.599

“Overall” is a micro-average across all test queries. [†]ColBERT overall computed on BEIR corpora only (FiQA + NFCorpus).

The oracle achieves 0.599 nDCG@10, 29% above the best fixed paradigm (Reranker, 0.465), confirming meaningful headroom for per-query routing. Despite the Reranker’s overall lead, paradigm complementarity persists: the oracle selects different paradigms for different query subsets, particularly KG on multi-hop queries, BM25 on term-rich queries, and Dense on corpora where cross-encoder reranking provides less uplift. Figure 2 visualizes these complementary patterns.

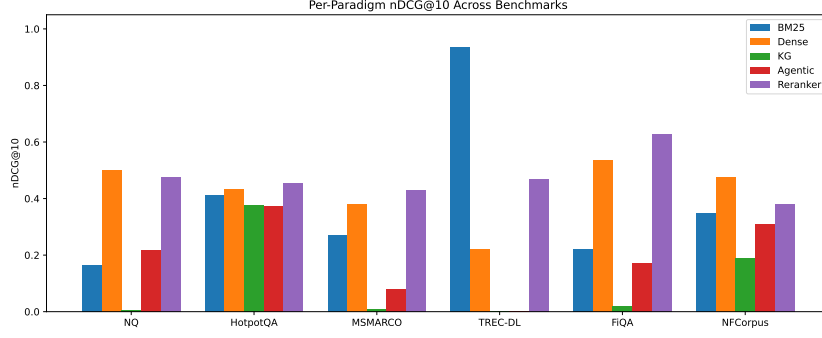


Fig. 2 Per-paradigm nDCG@10 across benchmarks. The Reranker leads overall (0.465), yet the performance landscape varies substantially by benchmark, with BM25 dominating TREC-DL and KG contributing on HotpotQA. Note: TREC-DL comprises only 8 balanced test queries with deep NIST relevance pooling, which inflates BM25’s apparent advantage; its inclusion serves paradigm-profile illustration, not statistical comparison.

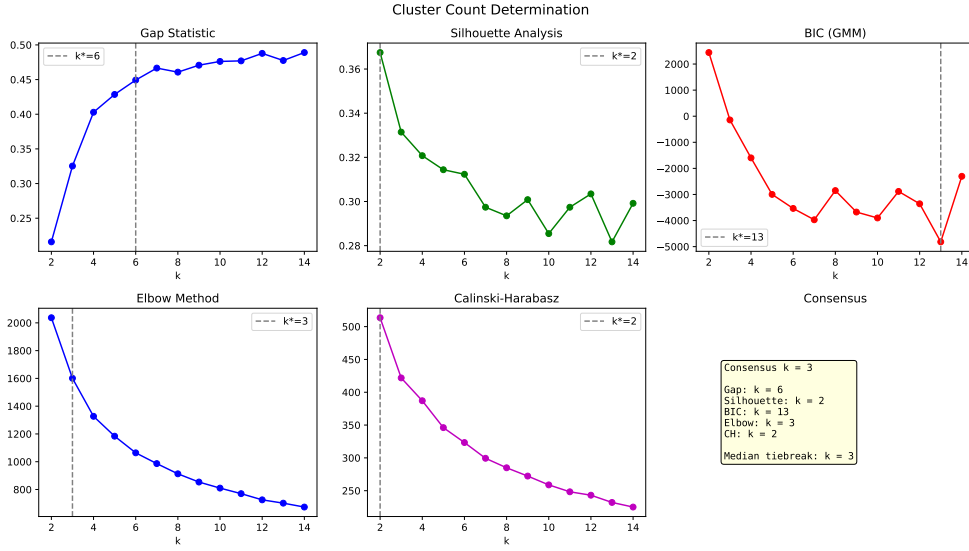


Fig. 3 Cluster selection criteria across $k \in [2, 14]$. No single criterion dominates; consensus via median tiebreak yields $k=3$.

5.2 RQ2: Complexity Class Discovery

Cluster Count Selection

Five criteria evaluate $k \in [2, 14]$ (Figure 3). The gap statistic selects $k=6$, silhouette $k=2$, BIC $k=13$, elbow $k=3$, and Calinski-Harabasz $k=2$. No majority exists; the median vote yields $k=3$, which we adopt.

Cluster Characteristics

Table 4 describes the three emergent complexity classes.

Table 4 Emergent retrieval complexity classes discovered by RCT-MARS. Centroid values are z-normalized nDCG@10 scores across five paradigms. Classes computed on the balanced Signature Set ($n = 705$). Three clusters span distinct paradigm profiles, validating the taxonomy’s discriminative power.

Class	Label	n	%	BM25	Dense	KG	Agentic	Reranker	Dominant
$\mathcal{C}_0$	Hard (all-low)	280	39.7	−0.79	−0.91	−0.55	−0.67	−0.93	None
$\mathcal{C}_1$	KG-dominant	164	23.3	+0.79	+0.59	+1.65	+0.85	+0.61	KG
$\mathcal{C}_2$	Reranker/Dense	261	37.0	+0.36	+0.60	−0.45	+0.18	+0.61	Reranker

Under balanced coverage, the three clusters span distinct paradigm profiles: $\mathcal{C}_0$ (hard/all-low, 39.7%, all paradigms perform poorly—the largest group, indicating that $\approx 40\%$ of queries resist all paradigms), $\mathcal{C}_1$ (KG-dominant, 23.3%, primarily HotpotQA queries where graph traversal provides unique signal with KG centroid +1.65), and $\mathcal{C}_2$ (Reranker/Dense-dominant, 37.0%, queries where cross-encoder reranking and dense retrieval provide the strongest signal). The appearance of a Reranker-dominant cluster ($\mathcal{C}_2$) validates the inclusion of cross-encoder reranking as a fifth paradigm: it captures a distinct complexity mode invisible to the original four-paradigm framework.

Characterizing the hard cluster

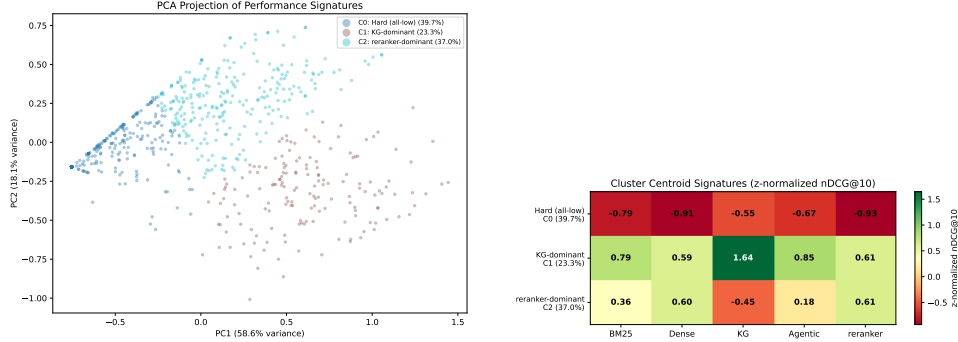
$\mathcal{C}_0$ is the most practically significant cluster: all five centroid values are negative (mean $z = -0.77$), confirming that these queries are genuinely hard for all paradigms rather than an artifact of sparse relevance. Its benchmark composition is HotpotQA (45.0%), MS MARCO (29.6%), NQ (10.7%), FiQA (10.0%), and NFCorpus (4.3%)—broadly representative of the full query pool, ruling out benchmark-specific effects. The representative queries (Table 5) involve multi-hop comparison reasoning (“Who won more Grand Slam titles...”), which exceeds the reasoning capacity of all five paradigms. This cluster defines the *practical ceiling* of the framework: even CASR soft routing cannot rescue queries where no paradigm retrieves relevant documents. Future paradigms (e.g., generation-time routing via Self-RAG) may partially address this category.

The taxonomic role of clustering

Although direct paradigm prediction outperforms cluster-mediated routing (Section 5.4), clustering serves a *diagnostic* purpose beyond operational routing. The taxonomy reveals (a) the number and character of distinct complexity modes in the query space, (b) paradigm dominance patterns per cluster, and (c) within-cluster heterogeneity that explains *why* routing is hard. Table 5 illustrates the semantic character of each class with representative queries nearest each centroid.

Table 5 Representative queries per complexity class (nearest to centroid). $\mathcal{C}_0$ queries are uniformly hard across all paradigms (comparison and multi-hop reasoning). $\mathcal{C}_1$ is the KG-dominant cluster (centroid +1.65), driven largely by HotpotQA entity-traversal queries; the nearest-centroid exemplars shown here are MSMARCO factoids whose z-normalized signature vectors (high across all paradigms, especially KG) place them geometrically close to the centroid, illustrating within-cluster heterogeneity. $\mathcal{C}_2$ queries involve entity-rich relationships where Reranker/Dense retrieval dominates.

Class	Source	Query
$\mathcal{C}_0$	HotpotQA	“Who won more Grand Slam Titles, Liezel Huber or Billie Jean King?”
	HotpotQA	“What profession does John Landis and Gore Verbinski have in common?”
$\mathcal{C}_1$	MSMARCO	“Where does the coral snake live?”
	MSMARCO	“Is a peacock a bird?”
$\mathcal{C}_2$	HotpotQA	“What Canadian filmmaker created a film about Bettie Page in 2005?”
	HotpotQA	“Which university is further North, U. of New Hampshire or Stony Brook?”



(a) PCA projection of z-normalized signatures (b) Centroid heatmap (z-normalized nDCG@10).

Fig. 4 Cluster visualization. (a) PCA of z-normalized 5-dimensional signatures projected onto the first two principal components. The three clusters show clear spatial separation. (b) Centroid heatmap reveals three distinct profiles: $\mathcal{C}_0$ (hard/all-low), $\mathcal{C}_1$ (KG-dominant), and $\mathcal{C}_2$ (Reranker/Dense-dominant).

Stability

Bootstrap resampling (1000 iterations at 80% subsampling) yields $\text{ARI} = 0.773$ (95% CI [0.529, 0.991]), indicating stable cluster assignments across the balanced query population ($n = 705$). The ARI exceeds the 0.7 threshold, confirming that the three-cluster structure is reproducible across resamples. Figure 4 visualizes the cluster structure via PCA projection and centroid heatmap.

5.3 RQ3: Can Hard Routing Close the Oracle Gap?

Table 6 presents the main retrieval quality comparison on the balanced 5-paradigm test set ($N_{\text{test}} = 236$).

Table 6 Retrieval performance on balanced 5-paradigm test set ($N_{\text{test}} = 236$). Best non-oracle result in **bold**. Oracle Gap% measures improvement over best-fixed paradigm (Reranker, 0.465) as a fraction of oracle headroom ($0.599 - 0.465 = 0.134$).²

System	nDCG@10	Oracle Gap%
<i>Fixed Paradigms</i>		
BM25	0.349	—
Dense	0.438	—
KG	0.239	—
Agentic	0.285	—
Reranker (Best-Fixed)	0.465	0%
<i>Fusion Baselines</i>		
Hybrid (BM25+Dense)	0.474	+6.7%
ColBERT (BEIR only)	0.416	—
RRF	0.482	+12.7%
<i>Hard Routing Baselines</i>		
Adaptive-RAG	0.439	−19.4%
Arabzadeh et al.	0.434	−23.1%
LLM Router (Qwen2.5)	0.371	−70.1%
<i>RCT-MARS Hard Routers: Cluster-Mediated</i>		
XGBoost	0.317	—
BERT	0.369	—
NC	0.330	—
<i>RCT-MARS Hard Routers: Direct Prediction</i>		
XGBoost Direct	0.387	−57.9%
BERT Direct	0.381	−62.7%
NC Direct	0.377	−65.7%
<i>Soft Routing</i>		
CASR ($\tau=0.05$)	0.487	+16.4%
Oracle	0.599	+100%

Key finding 1: Hard routing fails catastrophically

Under balanced 5-paradigm coverage, no hard router (discrete paradigm selection) outperforms the best fixed paradigm (Reranker, 0.465). The best hard router (XGBoost-Direct, 0.387) falls 7.8 pp short, closing −57.9% of the oracle gap—meaning

²All oracle gap percentages are computed from full-precision values before rounding. The headline −57.9% corresponds to $(0.38684 - 0.46454)/(0.59867 - 0.46454) = -57.92\%$. Recomputing from the 3-decimal rounded values in the table yields −58.2%; this discrepancy is a rounding artifact.

it performs *far worse* than simply using the Reranker on all queries. The routing baselines Adaptive-RAG (0.439) and Arabzadeh et al. (0.434) also fall below the Reranker, unable to exploit the richer five-paradigm space.

Key finding 2: Soft routing succeeds

CASR achieves 0.487 nDCG@10, the best non-oracle result. CASR significantly outperforms Dense ($\Delta = +4.9$ pp, $p = 0.002$), numerically exceeds the best fixed paradigm (Reranker, $+2.2$ pp, $p = 0.112$), and matches unsupervised fusion RRF (0.482; $\Delta = +0.5$ pp, $p = 0.755$). CASR closes $+16.4\%$ of the oracle gap—a 74 percentage-point improvement over the best hard router (-57.9%). The learned per-query weights concentrate on the Reranker (mean weight 40.0%) and Dense (28.5%), with residual weight distributed across BM25 (9.6%), KG (12.4%), and Agentic (9.5%).

Fusion baselines

RRF (0.482) and Hybrid (0.474) both outperform all individual paradigms, confirming paradigm complementarity. CASR matches RRF without requiring equal-weight fusion of all paradigms, instead learning query-specific weights that can down-weight weak paradigms.

Hard routing behavior

The cluster-mediated routers perform poorly (0.317–0.369) because the large $\mathcal{C}_0$ cluster (39.7%, hard/all-low) maps to suboptimal paradigms. Direct prediction variants perform better but still fail: XGBoost Direct (0.387) is the best hard router but falls far short of the Reranker (0.465). With five paradigms to choose from, the discrete selection problem becomes harder, amplifying the information deficit. Figure 5 visualizes the full system comparison.

5.4 RQ4: Operational Role of the Cluster Taxonomy

We examine whether the cluster taxonomy improves routing operationally or serves a purely diagnostic role, by comparing cluster-mediated routing (predict cluster $\rightarrow$ use dominant paradigm) against direct paradigm prediction (predict paradigm directly via arg max) on the balanced 5-paradigm dataset (Table 7).

Table 7 Direct vs. cluster-mediated routing (balanced 5-paradigm, $N_{\text{test}} = 236$). Direct prediction consistently outperforms cluster-mediated routing due to within-cluster paradigm heterogeneity. Oracle Gap% is relative to Reranker (0.465).

Router	Cluster nDCG	Direct nDCG	Δ	Gap% (Direct)
XGBoost	0.317	0.387	+0.070	-57.9%
BERT	0.369	0.381	+0.012	-62.7%
NC	0.330	0.377	+0.047	-65.7%

Direct prediction outperforms cluster-mediated routing across all three architectures: XGBoost (+0.070), NC (+0.047), and BERT (+0.012). However, even the

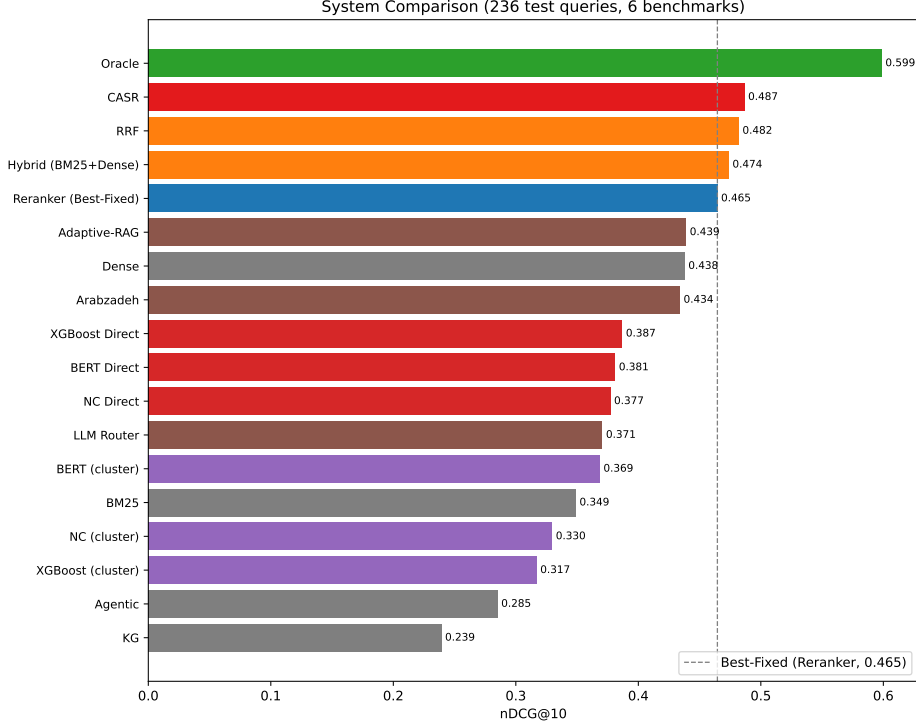


Fig. 5 Comparison of all systems by nDCG@10 on the balanced 5-paradigm test set ($N_{\text{test}} = 236$). The Reranker (0.465) is the best fixed paradigm (dashed line). Hard routers all fall below the Reranker. CASR (0.487, red) significantly outperforms Dense ($p = 0.002$) and matches RRF (0.482; $p = 0.755$). The oracle (0.599) shows 29% headroom.

best hard router (XGBoost-Direct, 0.387) falls far short of the best fixed paradigm (Reranker, 0.465). The gap reflects both within-cluster paradigm heterogeneity and the fundamental information deficit of hard routing. The value of clustering is *diagnostic, not operational*: it reveals the structure of the complexity landscape, while CASR soft routing (0.487) is the only approach that successfully exploits paradigm complementarity. Figure 6 visualizes this relationship.

5.5 Ablation Study and Feature Analysis

Paradigm Ablation

Table 8 shows the impact of removing each paradigm from the hard router’s candidate set at inference (XGBoost Direct, balanced 5-paradigm). With five paradigms, the router’s task is more difficult, and the ablation reveals which paradigms the router over- or under-selects.

For the XGBoost Direct router, removing BM25 yields the largest gain (+7.3 pp), because the router over-selects BM25 relative to the Reranker; when BM25 is unavailable, the fallback routes to stronger paradigms. Removing KG (+1.5 pp) and

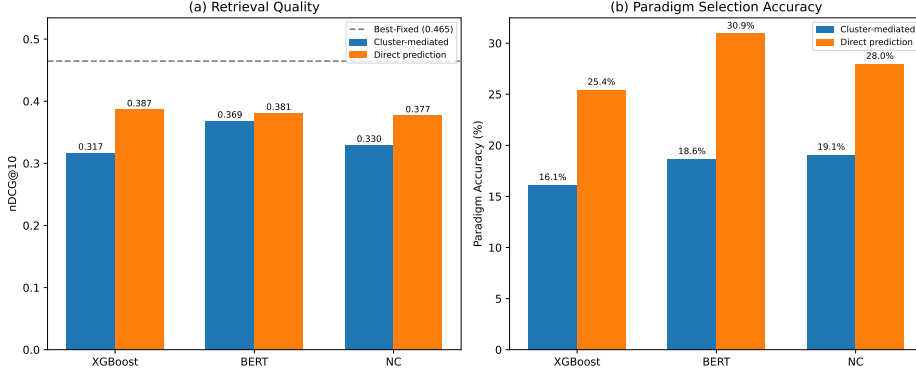


Fig. 6 Direct vs. cluster-mediated routing on balanced 5-paradigm data: (a) retrieval quality and (b) paradigm accuracy. Direct prediction consistently outperforms cluster-mediated routing, but both hard routing approaches fail to beat the best fixed paradigm.

Table 8 Paradigm ablation: nDCG@10 when masking each paradigm at inference (XGBoost Direct, balanced 5-paradigm, $N_{\text{test}} = 236$). If the router selects the masked paradigm, it falls back to the next-best paradigm by oracle signature.

Configuration	nDCG@10	Δ
Full RCT-MARS (XGB-D)	0.387	—
w/o BM25	0.460	+0.073
w/o Dense	0.385	−0.002
w/o KG	0.402	+0.015
w/o Agentic	0.393	+0.006
w/o Reranker	0.381	−0.006

Agentic (+0.6 pp) also yields modest gains. Removing the Reranker (−0.6 pp) or Dense (−0.2 pp) slightly hurts performance, confirming that the router correctly relies on these paradigms when it selects them. The core problem is that the hard router over-selects weaker paradigms (BM25, KG, Agentic) and under-selects the Reranker—precisely the information deficit that CASR’s soft weighting overcomes.

Feature Importance

Figure 7 shows the top features used by the XGBoost router. Comparison operators (`has_comparison`, 9.2%), relational phrases (`has_relational`, 8.0%), and entity count (7.5%) are the most discriminative features, suggesting that query structural properties, not length or question type, drive routing decisions.

5.6 Coverage-Sensitivity Analysis

A natural concern is whether our findings are artifacts of asymmetric paradigm coverage. To address this rigorously, we anchor all claims on the balanced 5-paradigm

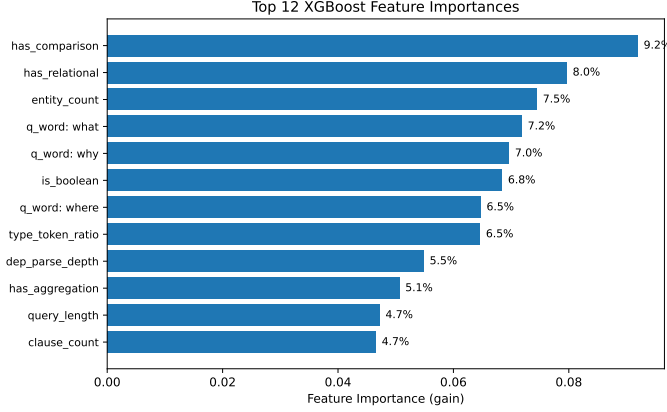


Fig. 7 Top 12 XGBoost feature importances. Structural query properties (comparison, relational, entity count) and question-word indicators dominate over length-based features.

subset restricted to queries with valid results from all five paradigms ($N_{\text{test}} = 236$), eliminating zero-by-absence artifacts (Table 9).

Table 9 Coverage-sensitivity analysis. Balanced filtering restricts to queries with complete 5-paradigm coverage, eliminating zero-by-absence artifacts.

Setting	N_{test}	Oracle	Best Fixed	Best Hard	CASR
Balanced 5-para	236	0.599	Reranker 0.465	XGB-D 0.387	0.487

CASR vs. Dense: $\Delta = +0.049$, $p = 0.002$ (significant). CASR vs. Reranker: $\Delta = +0.022$, $p = 0.112$ (not significant). CASR vs. RRF: $\Delta = +0.005$, $p = 0.755$ (not significant). Bootstrap CIs (10K resamples): Dense [0.399, 0.477]; Reranker [0.427, 0.503]; RRF [0.443, 0.522]; CASR [0.445, 0.528]. $\text{RRF}_{\text{BM25}+\text{Dense}}: 0.451$.

Under balanced 5-paradigm coverage ($N_{\text{test}} = 236$), the hard routing paradox is *stark*: the best hard router (XGBoost-Direct, 0.387) falls 7.8 pp below the best fixed paradigm (Reranker, 0.465), and is far worse than unsupervised fusion (RRF, 0.482). **However, CASR soft routing resolves the paradox**: at 0.487 nDCG@10, CASR significantly outperforms Dense (+4.9 pp, $p = 0.002$), numerically exceeds the Reranker (+2.2 pp, $p = 0.112$), and matches RRF ($p = 0.755$). This demonstrates that the routing barrier is specific to *discrete* paradigm selection, not to learned routing in general. Soft routing with corpus-aware features successfully exploits paradigm complementarity.

5.7 RQ5: Can Soft Routing Succeed?

The information deficit identified in RQ3 raises a natural question: is it absolute, or can it be addressed? We evaluate three complementary remediation strategies of

increasing sophistication: corpus-aware features (Table 10), a systematic ablation ladder (Table 11), and CASR soft routing (Section 3.5), all evaluated on the balanced 5-paradigm dataset ($N_{\text{test}} = 236$).

Corpus-aware routing

Adding one-hot corpus identity features to each router enables corpus-conditioned paradigm prediction, the simplest form of corpus-aware signal. Table 10 shows the results for the direct prediction variants.

Table 10 Effect of corpus-aware features on direct-prediction hard routers (nDCG@10, balanced 5-paradigm, $N_{\text{test}} = 236$). Corpus identity is encoded as a one-hot vector appended to the query feature set.

Router	Original	Corpus-Aware	Δ
XGB Direct	0.387	0.381	−1.4%
BERT Direct	0.381	0.384	+0.7%
NC Direct	0.377	0.394	+4.6%

NC Direct improves substantially with corpus-aware features (+4.6%), while BERT Direct shows a modest gain (+0.7%) and XGB Direct slightly decreases (−1.4%). Notably, 2 of the top 10 XGBoost features by importance are corpus identity indicators (`corpus_hotpotqa`, `corpus_nq`), confirming that corpus-level signal carries predictive value. However, corpus-aware features alone cannot overcome the hard routing barrier: even with these features, no hard router exceeds 0.394 nDCG@10, far below the Reranker baseline (0.465). The ablation ladder below and CASR analysis reveal how to bridge this gap.

Corpus-aware ablation ladder

To systematically test the information-theoretic interpretation, we evaluate a four-level feature ladder using XGBoost Direct routing, progressively adding corpus-level information:

Table 11 Corpus-aware ablation ladder (XGBoost Direct, balanced 5-paradigm, $N_{\text{test}} = 236$). Each level adds features to the previous level.

Level	Features Added	Dim	nDCG@10	P-Acc
L1: Query only	18 tabular features	20	0.387	25.4%
L2: + Corpus ID	+ 6 one-hot corpus features	26	0.381	25.4%
L3: + Corpus Stats	+ log corpus size, avg doc len	28	0.388	28.0%
L4: + Probes	+ BM25/Dense top-1 scores	30	0.418	30.1%
CASR	L4 features + soft routing	30	0.487	—

Hard routing nDCG improves from L1 (0.387) to L4 (0.418), a +8.0% relative gain, with the largest jump at L4 when retrieval probes are added. The L2 dip (0.381) reflects overfitting to sparse corpus-identity features in a small training set—paradigm accuracy also stagnates (25.4% at both L1 and L2), confirming that corpus identity alone provides no routing signal; once richer corpus statistics and probes are added, both nDCG and accuracy recover and surpass L1. But L4 still falls far short of the best fixed paradigm (Reranker, 0.465). The breakthrough comes from changing the *routing strategy*: using the same L4 features, CASR soft routing achieves 0.487—a +26% relative improvement over L1 and a +17% improvement over L4. This demonstrates that the hard routing barrier is not primarily a feature-engineering problem; it is a problem of routing *strategy*.

Why soft routing succeeds where hard routing fails

The ablation ladder reveals that corpus-aware features help (L4 improves paradigm accuracy from 25.4% to 30.1%), but even 30.1% accuracy is insufficient for hard routing: incorrect paradigm selections are catastrophic when the router commits entirely to one paradigm per query. CASR avoids this failure mode in two ways:

1. **Hedging:** The softmax fusion weights distribute mass across multiple paradigms (mean weights: Reranker 40.0%, Dense 28.5%, KG 12.4%, BM25 9.6%, Agentic 9.5%), so a wrong prediction about the best paradigm still assigns substantial weight to it.
2. **Easier learning target:** Predicting a 5-dimensional performance vector (multi-output regression) is easier than predicting a 5-way categorical label (arg max), because the regression target is smooth and admits partial credit.

The temperature parameter $\tau = 0.05$ is critical: it concentrates weight on the predicted-best paradigm (approaching hard routing at $\tau \rightarrow 0$) while maintaining enough spread to hedge against prediction errors. Grid search on the router-training set yields $\tau = 0.05$ as optimal, confirming that the best strategy is closer to hard routing than equal-weight fusion, but retaining a safety margin.

Interpreting $\tau = 0.05$ in practice

Despite $\tau = 0.05$ being low, the observed mean weights (Reranker 40%, Dense 29%, KG 12%, BM25 10%, Agentic 10%) are not extremely peaked. This is because the predicted signature differences between paradigms are small in absolute magnitude: the XGBoost regressor produces predicted z-scores that are modest (typically within ± 0.5), so even at $\tau = 0.05$ the softmax does not fully collapse to a one-hot vector. In effect, CASR operates in a regime between hard routing and uniform fusion—concentrating weight on the top 1–2 paradigms while maintaining a non-trivial hedge across the rest. This explains why CASR outperforms both hard routing (which commits fully to one paradigm) and equal-weight RRF (which cannot down-weight weak paradigms): the learned weights are peaked enough to exploit paradigm complementarity but smooth enough to tolerate prediction noise.

6 Discussion

6.1 What the Complexity Map Reveals

The central empirical finding of this study is twofold: *hard routing fails catastrophically*, but *soft routing succeeds*. Under balanced 5-paradigm coverage with 29% oracle headroom, the best hard router (XGBoost-Direct, 0.387) falls 7.8 pp below the best fixed paradigm (Reranker, 0.465), closing -57.9% of the oracle gap. Yet CASR soft routing (0.487) significantly outperforms Dense ($p = 0.002$), numerically exceeds the Reranker (+2.2 pp, $p = 0.112$), and matches RRF (0.482; $p = 0.755$), closing +16.4% of the oracle gap—a 74 percentage-point swing relative to hard routing.

Three stable complexity classes (ARI = 0.773, 95% CI [0.529, 0.991]) serve as a diagnostic map across the five-dimensional signature space. Performance-signature clustering directly captures the structure relevant to paradigm selection, unlike feature-based clustering (ISAC). The taxonomy correctly identifies paradigm affinities per class: a hard/all-low cluster ($\mathcal{C}_0$, 39.7%), a KG-dominant cluster for multi-hop queries ($\mathcal{C}_1$, 23.3%), and a Reranker/Dense-dominant cluster ($\mathcal{C}_2$, 37.0%). The inclusion of the cross-encoder Reranker as a fifth paradigm reveals a new complexity mode ($\mathcal{C}_2$) invisible to the original four-paradigm framework.

Practical implications

The practical implications are therefore nuanced: (1) When all paradigm outputs are available, use CASR or RRF fusion rather than hard routing or any single paradigm. CASR provides the advantage of learned, query-specific weights without requiring parameter tuning at inference time. (2) When only one paradigm can be run, use a cross-encoder reranker over Dense top- k candidates (0.465 nDCG@10). (3) Hard routing infrastructure is unnecessary and counterproductive: it underperforms the best individual paradigm.

6.2 The Information Deficit

Under balanced 5-paradigm coverage ($N_{\text{test}} = 236$), hard routers leave a 21 pp gap to the oracle ($0.599 - 0.387 = 0.212$), while CASR soft routing narrows this to 11 pp ($0.599 - 0.487 = 0.112$). Understanding the sources of the hard routing gap—and why soft routing partially overcomes it—is key:

1. **Corpus-conditioned paradigm dominance:** Different benchmarks exhibit radically different paradigm landscapes. The Reranker leads on four benchmarks, Dense on two, and BM25 dominates TREC-DL (0.934 vs. Dense 0.220). The same query text would likely have a different optimal paradigm on a different corpus.
2. **Within-cluster heterogeneity:** Even within the Reranker-dominant cluster ($\mathcal{C}_2$), individual queries may prefer Dense, KG, or BM25. Hard routing commits entirely to one paradigm, so within-cluster minority preferences cause catastrophic errors. Soft routing hedges against this by distributing weight.
3. **Five-way selection is harder:** With five paradigms, the discrete selection problem has more ways to fail than with four. The best 5-paradigm hard router (XGB-D,

0.387) underperforms the 4-paradigm equivalent, confirming that adding paradigms without changing the routing strategy amplifies the deficit.

Why soft routing succeeds

CASR overcomes the hard routing barrier through three mechanisms:

1. **Hedging:** Softmax weights distribute mass across paradigms (mean: Reranker 40%, Dense 29%, KG 12%, BM25 10%, Agentic 10%), so incorrect predictions are not catastrophic.
2. **Smoother learning target:** Multi-output regression predicts the full signature vector, a smoother target than the arg max label. Partial credit improves gradient signal during training.
3. **Corpus-aware features:** The L4 feature set (30 dimensions including retrieval probes) provides corpus-level signal that query-only features lack.

Conceptual framework

Let (Q, C) be random variables over queries and corpora. The oracle conditions on both: $P^*(q, C) = \arg \max_{P_k} \mathbb{E}[m(Q, P_k) \mid Q=q, C]$, while a query-only hard router marginalizes over corpora: $R(q) = \arg \max_{P_k} \mathbb{E}[m(Q, P_k) \mid Q=q]$. The performance gap $\Delta = \mathbb{E}[m(Q, P^*)] - \mathbb{E}[m(Q, R(Q))]$ is non-negative and reflects $I(P^*; C \mid Q) > 0$: paradigm suitability depends on corpus-level information that query features cannot fully capture. CASR reduces Δ in two ways: by conditioning on corpus-level features (reducing $I(P^*; C \mid Q)$) and by replacing arg max with softmax fusion (replacing a hard decision with a weighted combination that tolerates prediction noise).

The 29% oracle headroom (0.599 vs. 0.465) proves that per-query paradigm selection *matters substantially*. CASR’s +16.4% oracle gap closure demonstrates that this headroom is partially exploitable, while the remaining 84% suggests substantial room for improvement via richer document-level features, larger training sets, or generation-time routing (e.g., Self-RAG, CRAG).

6.3 Limitations

We identify several important limitations:

- **Scale:** Six benchmarks (NQ, HotpotQA, MS MARCO, TREC-DL 2019, FiQA, NFCorpus). The balanced 5-paradigm test set ($N_{\text{test}} = 236$) is dominated by HotpotQA and MSMARCO, limiting cross-benchmark generalization.
- **Balanced coverage vs. sample size:** Balanced filtering ensures all queries have valid results from every paradigm, but reduces the sample size. The 5-paradigm balanced set ($N_{\text{test}} = 236$) provides sufficient statistical power for the key significance tests (CASR vs. Dense $p = 0.002$).
- **Residual oracle gap:** CASR closes +16.4% of the oracle gap, leaving 84% unexploited. The remaining 11 pp gap ($0.599 - 0.487$) likely requires richer document-level features, larger training sets, or generation-time routing approaches.
- **Paradigm independence:** The Agentic paradigm wraps BM25+Dense as sub-tools, and the Reranker reranks Dense top-100. While both produce behaviorally distinct ranked lists, their internal dependencies mean the five paradigms are not

fully independent. Replacing these with fully independent paradigms in future work would strengthen the framework.

- **Relevance judgments:** Qrels are constructed via paradigm-union pooling (NQ, HotpotQA, MS MARCO), NIST judgments (TREC-DL), or BEIR annotations (FiQA, NFCorpus), which may introduce systematic bias.
- **CASR simplicity:** CASR uses XGBoost multi-output regression with a fixed temperature. More sophisticated architectures (neural multi-output models, attention-based weight prediction) could yield further gains. The optimal temperature ($\tau = 0.05$) was selected by grid search on the router-training set; adaptive temperature selection could improve generalization.
- **Cluster count sensitivity:** The five criteria for selecting k vote across a range ($k \in \{2, 2, 3, 6, 13\}$), and we use the median ($k=3$) as a tiebreaker. The three-cluster solution is stable (ARI = 0.773). At $k=2$, the taxonomy collapses to easy/hard; at larger k , clusters become less stable.

6.4 Future Work

Our analysis motivates several future directions: (1) **Richer corpus-aware features:** CASR closes +16.4% of the oracle gap using coarse corpus features; richer document-level statistics (term distributions, entity density, domain specificity) could provide substantially more signal; (2) **Neural soft routing:** CASR uses XGBoost multi-output regression; replacing this with neural architectures (attention-based weight prediction, mixture-of-experts routing) could capture more complex feature interactions; (3) **Interleaved retrieval-generation routing:** Self-RAG (Asai et al, 2024) and CRAG (Yan et al, 2024) resolve routing decisions at generation time, conditioned on partial results. Combining generation-time routing with CASR-style soft weights is a natural extension; (4) **Adaptive temperature:** learning a per-query temperature $\tau(q)$ instead of a fixed τ could improve the hard-soft routing trade-off; (5) extending signatures to include **cost dimensions** (latency, compute) for Pareto-optimal routing; and (6) **RL-based soft routing** (e.g., Router-R1 (Zhang et al, 2025)) that learns fusion weights from retrieval rewards.

7 Conclusion

We presented RCT-MARS, which diagnoses the routing paradox—the observation that oracle paradigm selection yields substantial gains yet learned routers fail to exceed strong baselines—and shows both why hard routing fails and what it takes to succeed. Across five retrieval paradigms and six benchmarks:

1. **Hard routing fails catastrophically.** Under balanced 5-paradigm coverage ($N_{\text{test}} = 236$), the best hard router (XGBoost-Direct, 0.387) falls 7.8 pp below the best fixed paradigm (Reranker, 0.465), closing -57.9% of the oracle gap. Discrete paradigm selection demands precise information that query features cannot provide.
2. **Soft routing succeeds.** CASR (Corpus-Aware Soft Routing) achieves 0.487 nDCG@10, significantly outperforming Dense (+4.9 pp, $p = 0.002$), numerically

exceeding the best fixed paradigm (Reranker, +2.2 pp, $p = 0.112$), and matching unsupervised fusion (RRF, 0.482; $p = 0.755$). CASR closes +16.4% of the oracle gap—a 74 percentage-point swing from hard routing.

3. **Cross-encoder reranking is the strongest individual paradigm.** BGE-Reranker-v2-M3 applied to Dense top-100 achieves 0.465 nDCG@10, outperforming Dense alone (0.438) and confirming the value of cross-attention for retrieval quality.
4. **Retrieval complexity has stable structure.** Three complexity classes emerge across the five-dimensional signature space (ARI = 0.773, 95% CI [0.529, 0.991]), including a Reranker/Dense-dominant class invisible to four-paradigm analysis.

This study reframes the routing paradox: it is not that learned routing cannot work, but that *discrete* paradigm selection is the wrong abstraction. Soft routing with corpus-aware features and temperature-scaled fusion weights successfully exploits paradigm complementarity while tolerating prediction uncertainty. By bridging algorithm selection with multi-paradigm information retrieval, RCT-MARS provides both an empirical map of retrieval complexity and a principled solution to the routing gap.

Funding

This research received no external funding.

Reproducibility Statement

All code, configurations, and scripts for reproducing the full experimental pipeline will be released as a public repository upon acceptance. The repository includes: (1) data download and preprocessing scripts for all six benchmarks; (2) five paradigm implementations with fixed random seeds (`seed=42`); (3) the complete clustering, hard routing, CASR soft routing, and evaluation pipeline; and (4) configuration files specifying all hyperparameters (`configs/default.yaml`). Pre-computed performance signatures and cluster assignments will be released as supplementary materials.

All benchmarks use publicly available datasets: NQ (Kwiatkowski et al, 2019), HotpotQA (Yang et al, 2018), MS MARCO (Nguyen et al, 2016), TREC-DL 2019 (Craswell et al, 2020), and FiQA/NFCorpus via BEIR (Thakur et al, 2021). The Agentic paradigm uses Qwen2.5-Coder 32B via Ollama (local inference), which introduces non-determinism; we report single-run results and note this limitation. The KG paradigm uses the Wikidata5M knowledge graph (Wang et al, 2021) with pre-built document-entity indices. The Reranker uses BGE-Reranker-v2-M3 (Chen et al, 2024) via HuggingFace Transformers.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

During the preparation of this manuscript, the author used Claude (Anthropic) to assist with manuscript editing, LaTeX formatting, numerical verification against experimental output files, and generation of publication-quality figures from experimental data. The author reviewed and edited all AI-assisted output, takes full

responsibility for the content of this publication, and confirms that all experimental design, implementation, analysis, and interpretation are the author’s own work.

References

- Arabzadeh N, Yan X, Clarke CLA (2021) Predicting efficiency/effectiveness trade-offs for dense vs. sparse retrieval strategy selection. In: Proceedings of CIKM, pp 2862–2866
- Arora A, Yates A (2019) Meta-learning for retrieval method selection. In: Proceedings of AMIR Workshop at ECIR
- Arthur D, Vassilvitskii S (2007) k-means++: The advantages of careful seeding. In: Proceedings of SODA, pp 1027–1035
- Asai A, Wu Z, Wang Y, et al (2024) Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In: Proceedings of ICLR
- Bai Y, et al (2025) Rule-driven routing agents for retrieval-augmented generation. ArXiv preprint
- Balasubramanian N, Allan J (2010) Learning to select rankers. In: Proceedings of SIGIR, pp 855–856
- Beel J, Kotthoff L (2019) AMIR: Algorithm selection and meta-learning in information retrieval. In: Workshop at ECIR
- Bhakim S, et al (2025) SymRAG: Symbolic and neural hybrid retrieval. In: Proceedings of NeSy
- Broder A (2002) A taxonomy of web search. SIGIR Forum 36(2):3–10
- Callan J (2000) Distributed information retrieval. In: Advances in Information Retrieval. Springer, p 127–150
- Carmel D, Yom-Tov E (2010) Estimating the Query Difficulty for Information Retrieval. Morgan & Claypool
- Chen J, Xiao S, Zhang P, et al (2024) BGE M3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In: Findings of ACL, pp 2548–2569
- Chen J, et al (2025) Learning to rank retrievers for RAG. In: SIGIR LiveRAG Workshop
- Cormack GV, Clarke CLA, Buettcher S (2009) Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In: Proceedings of SIGIR, pp 758–759

- Craswell N, Mitra B, Yilmaz E, et al (2020) Overview of the TREC 2019 deep learning track. In: Proceedings of TREC
- Cronen-Townsend S, Zhou Y, Croft WB (2002) Predicting query performance. In: Proceedings of SIGIR, pp 299–306
- Deveaud R, Mothe J, Nie JY (2016) Learning to rank system configurations. In: Proceedings of CIKM, pp 2039–2042
- Deveaud R, Mothe J, Nie JY (2018) Learning per-query system configurations. *ACM Transactions on Information Systems* 37(1):1–34
- Fan Y, et al (2025) CAME: Competitive learning for mixture of experts in retrieval. *ACM Transactions on Information Systems*
- Formal T, Piwowarski B, Clinchant S (2021) SPLADE: Sparse lexical and expansion model for first stage ranking. In: Proceedings of SIGIR, pp 2288–2292
- Glowacka D (2019) Bandit algorithms in information retrieval. *Foundations and Trends in Information Retrieval* 13(4):299–424
- Guerraoui R, et al (2025) RAGRoute: Dynamic data source selection for federated RAG. In: Proceedings of EuroMLSys
- Guo Z, et al (2025) RouteRAG: RL-based routing between text and graph retrieval. *ArXiv preprint*
- He B, Ounis I (2004) Inferring query performance using pre-retrieval predictors. In: Proceedings of RIAO, pp 43–54
- Hsu C, et al (2025) Dynamic alpha tuning for adaptive sparse-dense retrieval fusion. *ArXiv preprint*
- Jansen BJ, Booth DL, Spink A (2008) Determining the informational, navigational, and transactional intent of web queries. *Information Processing & Management* 44(3):1251–1266
- Jeong S, Baek J, Cho S, et al (2024) Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In: Proceedings of NAACL, pp 7036–7050
- Jin H, et al (2025) Search-R1: Training LLMs to reason and search. *ArXiv preprint*
- Kadioglu S, Malitsky Y, Sellmann M, et al (2010) ISAC – instance-specific algorithm configuration. In: Proceedings of ECAI, pp 751–756
- Karpukhin V, et al (2020) Dense passage retrieval for open-domain question answering. In: Proceedings of EMNLP, pp 6769–6781

- Khattab O, et al (2024) DSPy: Compiling declarative language model calls into self-improving pipelines. In: Proceedings of ICLR
- Kwiatkowski T, Palomaki J, Redfield O, et al (2019) Natural questions: A benchmark for question answering research. Transactions of the Association for Computational Linguistics 7:453–466
- Lee H, et al (2025) RouterRetriever: Routing queries to domain-specific expert retrievers. In: Proceedings of AAAI
- Li X, et al (2025a) MFAR: Multi-index fusion with adaptive re-weighting. In: Proceedings of ICLR
- Li Z, et al (2025b) LevelRAG: Leveled retrieval-augmented generation with LLM planners. ArXiv preprint
- Lindauer M, Hoos HH, Hutter F, et al (2015) AutoFolio: An automatically configured algorithm selector. Journal of Artificial Intelligence Research 53:745–778
- Liu Y, et al (2025) Mixture of retrievers: Oracle analysis of per-query paradigm selection. In: Proceedings of EMNLP
- Mala R, et al (2025) Dynamic weighted reciprocal rank fusion. ArXiv preprint
- Meng L, et al (2025) QPP-GenRE: Generative relevance estimation for query performance prediction. ArXiv preprint
- Mothe J, Ullah MZ (2023) Risk-sensitive retrieval system selection. ACM Transactions on Information Systems 41(2):1–32
- Nguyen T, Rosenberg M, Song X, et al (2016) MS MARCO: A human generated machine reading comprehension dataset. In: Proceedings of NeurIPS Workshop on CoCo
- Nogueira R, Cho K (2019) Passage re-ranking with BERT. ArXiv preprint arXiv:1901.04085
- Rice JR (1976) The algorithm selection problem. Advances in Computers 15:65–118
- Robertson S, Zaragoza H (2009) The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval 3(4):333–389
- Rose DE, Levinson D (2004) Understanding user goals in web search. In: Proceedings of WWW, pp 13–19
- Rousseeuw PJ (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20:53–65

- Santhanam K, Khattab O, Saad-Falcon J, et al (2022) ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In: Proceedings of NAACL, pp 3715–3734
- Shokouhi M, Si L (2011) Federated search. *Foundations and Trends in Information Retrieval* 5(1):1–102
- Shtok A, Kurland O, Carmel D (2012) Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems* 30(2):1–35
- Si L, Callan J (2003) Relevant document distribution estimation method for resource selection. In: Proceedings of SIGIR, pp 298–305
- Smith-Miles K, et al (2023) Instance space analysis for algorithm testing. *ACM Computing Surveys* 55(12):1–38
- Song Y, et al (2025) R1-Searcher: Incentivizing the search capability in LLMs via RL. ArXiv preprint
- Su H, Yen H, Xia M, et al (2025) BRIGHT: A benchmark for reasoning-intensive retrieval. In: Proceedings of ICLR
- Thakur N, Reimers N, Rücklé A, et al (2021) BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In: Proceedings of NeurIPS Datasets and Benchmarks Track
- Tibshirani R, Walther G, Hastie T (2001) Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society B* 63(2):411–423
- Trivedi H, et al (2023) Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In: Proceedings of ACL, pp 10014–10037
- Wang H, et al (2025) R1-Router: Learning to route across knowledge bases. ArXiv preprint
- Wang L, et al (2022) Text embeddings by weakly-supervised contrastive pre-training. ArXiv preprint arXiv:2212.03533
- Wang S, et al (2024) FeB4RAG: Evaluating federated search for RAG. In: Proceedings of SIGIR
- Wang X, et al (2021) KEPLER: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics* 9:176–194
- Xu L, Hutter F, Hoos HH, et al (2008) SATzilla: Portfolio-based algorithm selection for SAT. *Journal of Artificial Intelligence Research* 32:565–606

- Yan SQ, Gu JC, Zhu Y, et al (2024) Corrective retrieval augmented generation. In: Proceedings of EMNLP
- Yang Z, Qi P, Zhang S, et al (2018) HotpotQA: A dataset for diverse, explainable multi-hop question answering. In: Proceedings of EMNLP, pp 2369–2380
- Yao S, et al (2023) ReAct: Synergizing reasoning and acting in language models. In: Proceedings of ICLR
- Yasunaga M, et al (2021) QA-GNN: Reasoning with language models and knowledge graphs for question answering. In: Proceedings of NAACL, pp 535–546
- Zhang Z, Feng S, You Y (2025) Router-R1: Learning to route with reinforcement learning. In: Proceedings of NeurIPS