LEARNING MARGINAL VIEWPOINT UTILITY FOR BUDGETED EVIDENCE SELECTION IN RAG

Anonymous authors
Paper under double-blind review

ABSTRACT

Retrieval-augmented generation over multi-viewpoint queries must preserve distinct answers and positions from the retrieved evidence while operating under a fixed context budget. We introduce LMVU, a lightweight ranker that learns to prioritize candidate passages by the additional viewpoint coverage they contribute to the selected context. Training uses viewpoint annotations on training examples from each benchmark to compare candidate passages added to the same context. At inference, LMVU repeatedly scores passages that fit the remaining context budget as the context grows, without accessing these annotations. We evaluate recovered viewpoint coverage in terms of breadth (RVC_1) and repeated support from distinct sources (RVC_2). Exact oracles measure the maximum coverage achievable from the same evidence pool within the same budget. Across four viewpoint-annotated benchmarks at a 1,024-token context budget, LMVU achieves the highest average coverage on both metrics. It exceeds the strongest non-oracle baseline average by 0.037 RVC_1 points and 0.033 RVC_2 points, and reaches the exact viewpoint-breadth ceiling on two benchmarks.

1 INTRODUCTION

Retrieval-augmented generation (RAG) uses a retriever to find relevant text passages and a generator to produce a response conditioned on this evidence (Lewis et al., 2020). Conventionally, dense passage retrieval ranks passages by their relevance to a query using learned text representations, prioritizing highly ranked passages for inclusion in the context (Karpukhin et al., 2020).

However, some queries require coverage of different answers or positions to provide a comprehensive basis for a response. We use *viewpoint* to refer to an answer category or position supported by the evidence for a query. An account of electoral participation, for example, should include responses reporting three outcomes: voted, did not vote, and ineligible. A context containing only voters' responses represents just one viewpoint, even if every selected passage is accurate and relevant. For each query, the available passages form a candidate pool from which the context is selected. Under a limited context budget, several highly relevant passages may support the same viewpoint while leaving others unrepresented. This leads to a question:

How can we cover different viewpoints within a limited context budget?

A natural way to make better use of the context budget is to favor relevant passages that add information beyond what has already been selected. Maximal marginal relevance (MMR) implements this idea by balancing query relevance against similarity to previously selected passages (Carbonell & Goldstein, 1998). This trade-off also informs recent RAG methods that diversify the selected context to improve information recall (Wang et al., 2025b). However, semantic diversity need not imply viewpoint coverage: differently worded passages may support the same viewpoint, while similar wording may express different viewpoints.

We introduce Learning Marginal Viewpoint Utility (**LMVU**), a lightweight linear ranker that guides the selection of passages for a context covering diverse viewpoints. It ranks candidates by their predicted marginal viewpoint utility: the additional coverage they would contribute to the passages already selected. We train LMVU using viewpoint annotations on training examples from each benchmark. Given the same selected context, these annotations identify which candidate passage

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

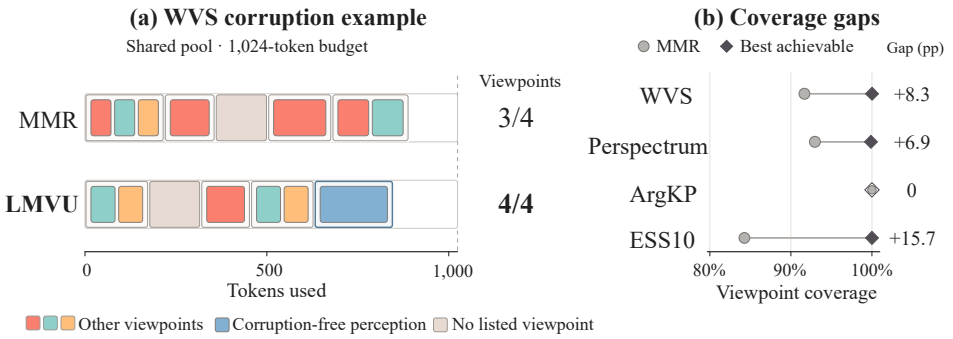


Figure 1: (a) Passages selected for a WVS corruption-perception query from the same pool under a 1,024-token budget. Bars show token use; colors indicate supported viewpoints, including repeated support, and beige denotes no match to the four annotated viewpoints. MMR covers three viewpoints; LMVU covers all four. (b) Mean viewpoint coverage of MMR and the exact full-pool optimum on each benchmark under the same budget; gaps are in percentage points.

adds more new viewpoints. The ranker learns these preferences using features including query relevance, passage length, and similarities to the selected passages. Starting from an empty context, LMVU scores candidates in the full query-specific pool that fit the remaining token budget, adds the highest-scoring passage, and updates the scores after each addition.

Figure 1(a) illustrates a World Values Survey (WVS) query asking how respondents differ in their perceptions of institutional corruption. From the same candidate pool and under a 1,024-token context budget, MMR repeatedly selects passages supporting three annotated viewpoints, leaving the corruption-free perception unrepresented. LMVU selects a passage supporting the missing corruption-free perception and covers all four viewpoints within the same 1,024-token budget.

We evaluate LMVU against relevance-based, diversity-aware, and learned selectors, including neural rerankers, on four benchmarks at a 1,024-token budget (Table 1). We measure recovered viewpoint coverage in terms of breadth (RVC_1) and repeated support from distinct sources (RVC_2). Figure 1(b) shows the gap between MMR's mean coverage and the maximum achievable from the same candidate pools under the same budget. LMVU achieves the highest average coverage on both metrics, with each benchmark weighted equally: it exceeds the strongest non-oracle baseline average for each metric by 0.037 and 0.033 points, respectively. It reaches full viewpoint coverage on WVS and ArgKP and attains 94.6% on Perspectrum (§5). In summary, our contributions are:

- We show that budget-constrained passage selection can leave supported viewpoints unavailable to downstream generation, even when the selected passages are relevant. Exact coverage oracles reveal how much additional viewpoint coverage is achievable from the same candidate pools.
- We introduce LMVU, a lightweight linear ranker trained to predict the additional viewpoint coverage a candidate passage contributes to a growing context. Its scores adapt to the passages already selected, helping allocate the remaining context budget.
- Across four benchmarks, LMVU achieves the highest average viewpoint breadth and repeated support among all evaluated selectors. Further experiments show that considering the selected context helps identify passages that add new viewpoints, and that training on other benchmarks can improve coverage on a new benchmark.

2 Related Work

Passage diversification. Maximal marginal relevance balances query relevance against similarity to selected passages (Carbonell & Goldstein, 1998). Other approaches diversify passages through facility-location objectives (Lin & Bilmes, 2011; Mirzasoleiman et al., 2015), explicit aspects (Santos et al., 2010), determinantal point processes (Kulesza & Taskar, 2012), or graph centrality (Erkan

& Radev, 2004). These methods use relevance, similarity, or proxy aspects to guide selection. Our objective is the additional coverage a passage contributes to the viewpoints supported by a query.

Budgeted and learned context selection. Recent selectors adapt relevance–redundancy scoring to the budget (Peng et al., 2025), vary the relevance–diversity trade-off by query (Khan et al., 2026), or constrain selection by passage clusters (Xu et al., 2026). JPR ranks passages jointly to cover multiple answers (Min et al., 2021), while RichRAG uses a generative listwise ranker to cover query aspects and account for the generator's preferences (Wang et al., 2025a). LMVU learns within-context preferences from annotated marginal viewpoint gains, fitting a linear scorer over frozen passage representations and updating scores as the context grows.

Multi-viewpoint benchmarks. We evaluate on survey responses from WVS and ESS10 (Zhao et al., 2024; European Social Survey European Research Infrastructure (ESS ERIC), 2023), claim perspectives from Perspectrum (Chen et al., 2019), and arguments from ArgKP (Bar-Haim et al., 2020). Related opinion and value resources examine whose views models express (Durmus et al., 2023; Sorensen et al., 2023; Kirk et al., 2024); our evaluation measures which supported viewpoints enter the selected context.

3 Problem Formulation

Budget-constrained viewpoint coverage. For a query q, let $E(q)$ be the full prequalified evidence pool and V_q its benchmark-annotated viewpoint inventory. Each candidate passage e has token cost $c(e)$ and supports viewpoints $V(e) \subseteq V_q$; we write $V(S) = \bigcup_{e \in S} V(e)$ for the viewpoints covered by a selection. The objective is to select a context $S \subseteq E(q)$ satisfying $\sum_{e \in S} c(e) \leq B$ that maximizes *recovered viewpoint coverage*:

$$\mathrm{RVC}(S) = \frac{\left|\bigcup_{e \in S} V(e)\right|}{|V_q|}.$$

Each viewpoint receives credit once, regardless of repeated support.

Label-blind selection. Gold viewpoint annotations provide training supervision and evaluation targets but are unavailable at inference. The selector must therefore estimate coverage gains using observable evidence features and the current selection state.

Exact oracle and reference headroom. With gold access, at a fixed budget B the full-pool oracle solves $S_q^\star \in \arg\max_{S \subseteq E(q)} \mathrm{RVC}(S)$ subject to $\sum_{e \in S} c(e) \leq B$; we compute this optimum by viewpoint-subset dynamic programming (Appendix E). For a reference allocator $\mathcal{A}$ producing a feasible selection $S_{\mathcal{A}}(q, B)$, the reference headroom $\mathrm{headroom}_{\mathcal{A}}(q, B) = \mathrm{RVC}(S_q^\star) - \mathrm{RVC}(S_{\mathcal{A}}(q, B))$ is the maximum coverage improvement achievable over $\mathcal{A}$ under the same evidence pool and budget. Benchmark-level headroom averages these gaps over evaluation queries.

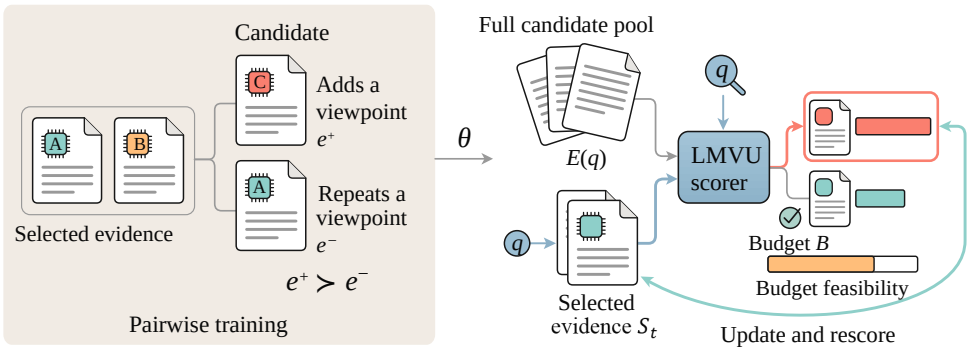
4 Method: Learned Marginal Viewpoint Utility

4.1 Marginal utility and training states

The coverage objective of The coverage objective of §3 defines the signal a runtime allocator seeks to estimate: the marginal gain $\Delta V(e \mid q, S)$ of a candidate given the current selection,

$$\Delta V(e \mid q, S) \;=\; \big|V(e) \setminus V(S)\big|.$$

The added coverage of the same candidate changes as the selection grows and falls to zero once its viewpoints are covered; because attainable gains depend on the current selection, training uses within-state comparisons that match the inference-time ranking decision. Computing ΔV requires gold viewpoint annotations, so the oracle is not deployable, but on development data the realized gains are measurable, and they supervise a surrogate. LMVU (*learned marginal viewpoint utility*) fits a scoring function $f_\theta(q, S, e)$ over runtime-visible, label-blind features to rank candidates by



(a) Learning marginal utility **(b) Selection under a token budget**

Figure 2: LMVU overview. (a) Learning marginal utility (development, gold visible),with A and B already covered, a candidate adding viewpoint C is the positive e^+, one repeating a covered viewpoint the negative e^-; realized gold gains give within-state preferences $e^+ \succ e^-$ for pairwise training, and only the learned θ leaves this stage. (b) Selection under a context budget (runtime, label blind), the scorer ranks budget-feasible candidates from the full pool $E(q)$; the top pick enters S_t, the state updates, and candidates are rescored until the budget is exhausted.

estimated marginal gain. It uses this ranking in a sequential budget-feasible allocator over the full evidence pool $E(q)$, rescoring candidates as the selection grows (Figure 2).

Training states. Development states are harvested on the frozen development pools $P(q) \subseteq E(q)$, λ=0.5 MMR-oversampled, indexed by budget, and used only to harvest training states, by rolling out five state-visitation policies: dense-relevance greedy, live MMR, two pinned-seed random orders, and a gold-guided marginal-gain policy. Prefixes (q, S_t) of up to nine steps are snapshotted, order-equivalent states are deduplicated per query–budget pair, and each state scores at most twenty label-blind candidates (dense top ten plus a deterministic uniform draw), with gold annotations giving realized gains $\Delta V(e \mid q, S_t)$; unequal-gain candidates within a state supply ranking pairs. Training is confined to the local pools $P(q)$, while the fitted scorer deploys on the full pools $E(q)$; §5 evaluates this transfer from local training states to full-pool selection across four benchmarks.

4.2 State-conditioned scoring and learning

f_θ is linear in an 18-dimensional feature vector $\phi(q, S, e)$,

$$f_\theta(q, S, e) \;=\; \theta^\top \tilde{\phi}(q, S, e),$$

where $\tilde{\phi}$ is ϕ standardized per dataset on that dataset's development records. The features fall into four groups, candidate descriptors (4), evidence–state geometry (5), atomic–state geometry (5), and shared state descriptors (4), defined in full in Appendix M (Table 8). The candidate descriptors retain relevance and length information, while the state features describe how each passage relates to the context already selected. All eighteen are computable at runtime without gold viewpoint fields, and all similarity geometry reuses the frozen embeddings already computed for the dense scorer, adding no further model dependency. The static ablation M0 restricts θ to three candidate descriptors: dense relevance, query–candidate cosine, and candidate token length. The four shared state descriptors are constant across candidates within a state, means they cancel in the within-state differences below and cannot alter any within-state argmax; their fitted coefficients are exactly zero in every fitted model. The remaining budget determines which passages are eligible through the feasibility filter of §4.3; LMVU then ranks those eligible passages using the learned score.

Pair construction. Pointwise regression of ΔV conflates across-state scale differences, realized gains range from zero to the number of as-yet-uncovered viewpoints, and the attainable scale shrinks as coverage accumulates. We therefore train by *within-state* pairwise ranking. For each development state (q, S) we form ordered pairs (e^+, e^-) with $\Delta V(e^+ \mid q, S) > \Delta V(e^- \mid q, S)$, never pairing candidates across states, capped at 20 pairs per state via deterministic subsample. Tied pairs carry no

signal and are excluded. Every retained pair contributes two antisymmetric examples, $(\tilde{\phi}^+ - \tilde{\phi}^-, 1)$ and $(\tilde{\phi}^- - \tilde{\phi}^+, 0)$, making the preference direction-consistent and an intercept unnecessary.

Objective. The surrogate is fit by a regularized pairwise logistic (Bradley–Terry) loss over standardized symmetric feature differences,

$$\min_{\theta} \ \frac{1}{2}\|\theta\|^2 \ + \ C \sum_{(q,S)} \sum_{(e^+,e^-)\in\mathcal{P}(q,S)} \log\Big(1+\exp\big(-\theta^\top\big(\tilde{\phi}(q,S,e^+)-\tilde{\phi}(q,S,e^-)\big)\big)\Big),$$

where $\mathcal{P}(q,S)$ is the state's sampled pair set. Two properties motivate this form. (i) Differencing cancels additive state-specific offsets and the per-dataset standardization mean, leaving only per-feature scale normalized and within-state orderings free of across-state gain scale. (ii) A pairwise loss optimizes exactly the decision the allocator faces, ranking the budget-feasible candidates of one state—and amounts to a concordance (AUC-type) criterion on realized gains. §5.3 reports exactly this criterion on development states (Table 1(b)). The objective is convex, optimized to convergence with LBFGS (C=1.0); through the logistic link, the model parameterizes the pairwise preference probability that e^+ adds more coverage than e^- in state (q,S).

Remark. MMR's fixed λ trade-off uses fixed coefficients to combine relevance with selection-dependent similarity at every step (Remark 8); in LMVU the learned weights are fixed at inference, but they act on geometry features that update with the selection, so the effective novelty penalty varies with what is already covered, the same candidate can rank highly early and drop once its viewpoint enters the selection (the rotation behavior visible in the case studies of Appendix L).

4.3 Sequential allocation

Algorithm 1 LMVU budget-feasible greedy allocation

Input: query q; full pool $E(q)$; budget B; trained scorer f_θ
Output: context $S \subseteq E(q)$ within budget B

$S_0 \leftarrow \emptyset$; $u_0 \leftarrow 0$; $t \leftarrow 0$
while true **do**
  $F_t \leftarrow \{\, e \in E(q) \setminus S_t \ : \ u_t + c(e) \le B \,\}$
  if $F_t = \emptyset$ **then break**
  $e^* \leftarrow \arg\max_{e\in F_t} f_\theta(q, S_t, e)$
  $S_{t+1} \leftarrow S_t \cup \{e^*\}$; $u_{t+1} \leftarrow u_t + c(e^*)$; $t \leftarrow t+1$
end while; **return** S_t

At inference, the allocator repeats, score budget-feasible candidates with $f_\theta(q, S_t, \cdot)$, add the argmax, update the state, and stop when no candidate fits or the pool is exhausted (Algorithm 1, Figure 2). Token accounting uses whole-evidence token lengths; partial evidence is never selected, and a candidate is eligible only while its full cost fits the remaining budget, never truncating an item to fit.

Cost and determinism. Scoring costs $O(|F_t|\, d)$ per step with d=18 features; the feature engine maintains incremental sufficient statistics over the selection, running similarity extrema, the selected-embedding sum, per-atom maxima, with no pool-level similarity matrix materialized. The argmax is deterministic; inference runs one deterministic pass per query (Appendix A).

Properties. LMVU inherits the greedy loop's feasibility guarantee but claims no coverage approximation bound: under the knapsack budget even raw-gain greedy carries no constant-factor guarantee, and the learned surrogate is not the true submodular gain (formal statements: Appendix E). Whether the predicted ordering tracks realized marginal gains is measured directly by the development-fold ranking metrics and learnability analysis (Appendices F and G).

5 Experiments and Analysis

5.1 Setup

Benchmarks. We evaluate four viewpoint-annotated benchmarks: survey-derived WVS, claim–perspective Perspectrum (Chen et al., 2019), argument-key-point ArgKP (a saturation control) (Bar-Haim et al., 2020), and ESS10 survey items (European Social Survey European Research Infrastructure (ESS ERIC), 2023). Selection uses the *full* pool $E(q)$ (mean sizes $507/9/76/88$ at budget 1024; Appendix N); training-state harvesting alone uses smaller frozen λ=0.5 MMR-oversampled pools ($4\times$ budget multiplier; §4). Budgets are $\{512, 1024, 2048\}$ (1024 primary). The protected WVS fold allows only predeclared evaluations and read-only audits, never fitting (Appendix A). Pool and feature geometry uses one frozen local all-MiniLM-L6-v2 encoder (Reimers & Gurevych, 2019).

Metrics. Let $n_v(S)$ count distinct source units supporting viewpoint v in S, survey respondents in WVS and ESS10, claim-scoped documents in Perspectrum, and individual arguments in ArgKP. For a positive integer n, we evaluate

$$\mathrm{RVC}_n(S) \;=\; \frac{1}{|V_q|}\sum_{v\in V_q}\min\{n_v(S)/n,\,1\}.$$

Here $n = 1$ recovers RVC (breadth); we also report $n = 2$ (RVC_2, repeated support), where zero, one, or at least two supporting units contribute 0, 0.5, or 1 per viewpoint. The offline gold-access bound $\mathrm{Oracle}_n(q, B)$ is the maximum $\mathrm{RVC}_n(S)$ over selections from the same full pool $E(q)$ satisfying $\sum_{e\in S} c(e) \le B$. Oracle denotes Oracle_1; Oracle_2 optimizes repeated support separately, so their selected subsets may differ (Appendix P).

Baselines. We compare Random, BM25, FacLoc, xQuAD, SubMix, LexRank, Dense, CeRerank, Cluster, MMR, static M0, and LMVU (M1), sharing full pools and token accounting. Four published methods are evaluated under the same pools, budgets, and protocol: the allocation heuristics AdaGReS (Peng et al., 2025) and the MMKP selector of Xu et al. (2026), and the training-heavy neural rerankers JPR (Min et al., 2021) and RichRAG (Wang et al., 2025a), whose rows are shaded separately in Table 1. The predeclared kDPP comparator appears in the appendix budget grid (Table 11); definitions and five predeclared baseline locks are in Appendix A. For development-state ranking, we report normalized discounted cumulative gain at rank five (NDCG@5), using realized marginal coverage gains as relevance labels, and pairwise concordance with the true gain ordering.

Protocol. Each benchmark follows adapter verification → frozen pools → grouped development training and lock → held-out evaluation. Audits found zero runtime gold-field or provider accesses (Appendix A). The LF matrix and full-pool baseline rerun contain 28,314 per-query allocator runs. Queries are the evaluation units: 6, 97, 6, and 34 for WVS, Perspectrum, ArgKP, and ESS10, respectively. WVS uses protected respondent fold USA 4 over the same queries; transfer benchmarks use development/held-out query splits. Deterministic allocators contribute one trajectory per query. Ten seeds vary only Random, xQuAD, and Cluster; learned rows (M0, JPR, RichRAG, LMVU) vary only through three training-seed refits, with deterministic inference. Given prior held-out inspection, this full-pool extension is exploratory rather than a fresh confirmatory test.

5.2 Coverage under fixed budgets

Table 1 reports coverage at the primary 1,024-token budget. LMVU achieves the highest average across the four benchmarks among all sixteen non-oracle methods, with 0.9588 RVC and 0.8795 RVC_2. The strongest competing averages are 0.9222 from MMR for breadth and 0.8469 from FacLoc for repeated support. Relative to MMR, the exact oracle leaves 0.083, 0.069, and 0.157 RVC points of headroom on WVS, Perspectrum, and ESS10, respectively (Table 9). LMVU reaches the exact breadth ceiling on WVS and ArgKP and improves over MMR by 0.0161 on Perspectrum and 0.0466 on ESS10, the benchmark with the largest measured headroom.

The adapted neural selectors have strong results on particular benchmarks. JPR reaches full ArgKP coverage at 512 tokens, while RichRAG obtains the highest ESS10 scores at 1,024 tokens, with 0.9569 RVC and 0.8315 RVC_2. LMVU retains the highest average on both metrics across the four benchmarks. Table 1(b) presents development ranking and transfer analyses, while panel (c) reports paired ESS10 differences against individual comparators.

Paired statistics are reported in Table 12 and Appendix F. The Perspectrum repeated-support gain over MMR is $+0.0213$ with a 95% bootstrap interval of $[+0.007, +0.037]$. On the largest benchmark, LMVU fits its linear scorer on cached features in 1.3 s on CPU. Adapting JPR and RichRAG takes 150.5 s and 962.7 s per fit on GPU, respectively; both require neural scoring during selection.

5.3 What does state conditioning contribute?

Table 1 contains the key ablation, M0, the static 3-feature scorer. State-dependence drives the three largest gains, LMVU beats M0 by $+0.202$ (WVS), $+0.075$ (ArgKP, where M0 drops below the otherwise universal ceiling at 0.9250), and $+0.061$ (ESS10). Perspectrum is the exception where M0 exceeds LMVU by 0.004, a benchmark-level boundary we analyze in §6. Dense top-k never

Table 1: Main results at budget 1024 on the full per-query pools.

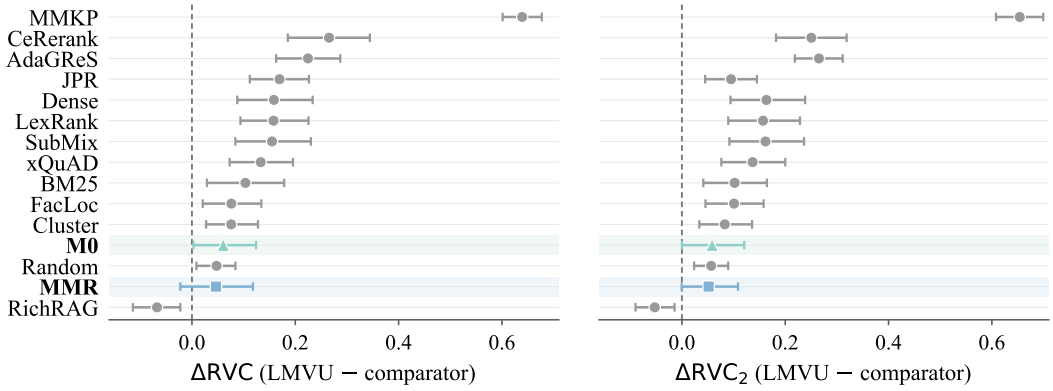
(a) Viewpoint breadth (RVC) and repeated support (RVC_2) at 1,024 tokens. Avg gives the equal-weight mean across benchmarks; oracle rows are exact full-pool bounds ('–' denotes inapplicable entries). Bold and underline mark the best and second-best non-neural methods. Shading is scaled by column; yellow neural rows are excluded from these highlights (Appendix A).

	RVC (breadth)					RVC_2 (repeated support)				
Method	WVS	Perspectrum	ArgKP	ESS10	Avg	WVS	Perspectrum	ArgKP	ESS10	Avg
Random	0.9088	0.9111	0.9933	0.8416	0.9137	0.8757	0.7932	0.9720	0.7223	0.8408
BM25	0.7000	0.8859	1.0000	0.7858	0.8429	0.6583	0.7720	0.9881	0.6769	0.7738
FacLoc	0.9167	0.9108	1.0000	0.8130	0.9101	**0.9167**	0.8045	0.9881	0.6781	0.8469
xQuAD	0.7700	0.9239	1.0000	0.7562	0.8625	0.7114	0.8083	0.9733	0.6422	0.7838
SubMix	0.7833	0.9139	1.0000	0.7343	0.8579	0.6898	0.8002	0.9714	0.6173	0.7697
LexRank	0.7833	0.9122	1.0000	0.7314	0.8567	0.7315	0.7986	0.9881	0.6220	0.7850
AdaGReS	0.7167	0.7509	0.9762	0.6647	0.7771	0.6148	0.5018	0.9143	0.5138	0.6362
MMKP	0.6241	0.9482	0.9667	0.2502	0.6973	0.3120	0.8329	0.9173	0.1251	0.5468
Dense	0.7833	0.9126	1.0000	0.7306	0.8566	0.6898	0.8014	0.9714	0.6154	0.7695
CeRerank	0.7380	0.9027	1.0000	0.6239	0.8161	0.6704	0.7923	0.9458	0.5285	0.7343
Cluster	0.8833	0.9135	0.9752	0.8134	0.8964	0.7753	0.7999	0.9631	0.6960	0.8086
MMR	0.9167	0.9296	1.0000	0.8427	0.9222	0.7824	0.8160	1.0000	0.7273	0.8314
M0	0.7981	**0.9495**	0.9250	0.8287	0.8753	0.7444	**0.8480**	0.9131	0.7206	0.8065
LMVU	**1.0000**	0.9457	**1.0000**	**0.8893**	**0.9588**	0.9015	0.8373	**1.0000**	**0.7790**	**0.8795**
JPR	0.8361	0.9074	1.0000	0.7197	0.8658	0.8097	0.7996	1.0000	0.6839	0.8233
RichRAG	0.8167	0.8908	0.9234	0.9569	0.8970	0.7994	0.7906	0.9234	0.8315	0.8362
Oracle	1.0000	0.9987	1.0000	1.0000	0.9997	–	–	–	–	–
$Oracle_2$	–	–	–	–	–	1.0000	0.8898	1.0000	0.9334	0.9558

(b) Development-only ranking (left) and leave-one-benchmark-out transfer at budget 1024 (right). Ranking scores average seeds and, for WVS, three folds; Δ is relative to MMR. Best per triple in **bold**; shading indicates magnitude.

	dev within-state NDCG@5			dev pairwise concordance			LOBO transfer (dev, 1024)	
Benchmark	LMVU	$M0^+$	M1-SHUF	LMVU	$M0^+$	M1-SHUF	RVC	Δ vs. MMR
WVS	**0.7944**	0.6326	0.6002	**0.8260**	0.7184	0.6892	1.0000	+0.1500
Perspectrum	0.7927	**0.8004**	0.7960	**0.5829**	0.5375	0.5328	0.9510	+0.0443
ArgKP	**0.4629**	0.3539	0.3496	**0.6380**	0.5266	0.5243	0.9885	0.0000
ESS10	**0.5715**	0.5498	0.5502	**0.6098**	0.5733	0.5736	0.8452	+0.0374

(c) Paired LMVU−comparator differences on held-out ESS10, means and 95% bootstrap CIs; same order in both halves on a shared scale.



exceeds MMR (tying only at the ArgKP ceiling) and stays far from the ceiling where headroom exists (ESS10 0.731 vs. 1.000 for the exact oracle), relevance alone does not solve the allocation.

Two development diagnostics complement these held-out results. In the pre-registered learnability probe (Appendix G), within-state ranking on development states puts M1 at NDCG@5 0.6575, ahead of the MMR-style heuristic (0.6313) and M0 (0.5869). Removing the six atomic-granularity features collapses WVS development ranking (NDCG@5 $0.6597 \rightarrow 0.4418$; top-1 positive-gain rate $0.5425 \rightarrow 0.4088$), leaving Perspectrum and ArgKP unchanged (Appendix H). Both are development-fold diagnostics that localize the state signal's source; neither is held-out evidence.

Table 1(b) (left) adds two predeclared controls on the same development states. A capacity-matched null that keeps all 18 features but shuffles the 14 state features within each state (M1-SHUF) collapses ranking to the static level (WVS 0.6002 vs. LMVU's 0.7944; ArgKP 0.3496 vs. 0.4629), so the ranking gain comes from state information rather than capacity. This drop is most visible on WVS, where the choice of a useful next passage depends strongly on which viewpoints the context already represents.Conversely, augmenting the static scorer with a fourth candidate descriptor (M0^{+}) leaves its selections identical to M0's on Perspectrum, ArgKP, and ESS10 and its WVS breadth at 1024 unchanged (0.7981), while its development ranking stays far below LMVU's exactly where the held-out gaps are large. On Perspectrum, both static variants slightly outrank LMVU on development states, consistent with the held-out LMVU–M0 difference of -0.004 RVC points.

5.4 Cross-benchmark transfer

If the learned coefficients capture a reusable coverage regularity rather than benchmark-specific annotation patterns, a model trained without any data from a benchmark should still allocate well on it. Table 1(b) (right) reports this leave-one-benchmark-out probe on development folds at budget 1024, one model trained on the pooled development states of the other three benchmarks improves over MMR on WVS ($+0.150$), Perspectrum ($+0.044$), and ESS10 ($+0.037$), ties at the ArgKP ceiling, and attains full breadth on WVS (1.0000). The positive deltas on three benchmarks show that the learned coefficients remain useful when fitted entirely outside the target benchmark. All four cells use development folds only, so the probe leaves the held-out firewall intact; the WVS cell in particular never touches the protected fold.

5.5 Breadth and repeated support

Under repeated-support coverage, the ordering largely carries over (M0 leads Perspectrum, LMVU leads ESS10, ArgKP remains tied at the ceiling), with one inversio that on WVS the relevance-free FacLoc edges LMVU (0.9167 vs. 0.9015), though both sit well below the depth oracle. LMVU's gains over MMR remain positive on all headroom-positive benchmarks ($+0.119$, $+0.021$, $+0.052$). Depth also holds the larger residual headroom,headroom$_2$ is 0.218 (WVS), 0.074 (Perspectrum), and 0.206 (ESS10), against 0.083, 0.069, and 0.157 for breadth (Table 9). The residual gaps differ, at budget 1024, LMVU's remaining distance to the full-pool oracle on WVS is 0.0000 in breadth but 0.0985 in depth, breadth is at the pool optimum and only repeated support is missing, while ESS10 retains a 0.1107 breadth and a 0.1544 depth gap.

5.6 Robustness across embedding models

Table 2 compares the canonical MiniLM-L6 pipeline (full-pool protocol) against two further bi-encoders (bge-base, bge-m3) under the earlier local-selection protocol. The gains depend on the embedding model. With bge-base, LMVU improves over MMR by 0.025 RVC on Perspectrum and 0.074 on ESS10, with bootstrap intervals excluding zero; bge-m3 yields smaller positive mean gains on both, with intervals crossing zero, and a negative mean difference on WVS (-0.0417). These evaluations rebuild embeddings, pools, and features together, characterizing pipeline sensitivity. Full scores and uncertainty estimates are in Appendix J.

5.7 Sensitivity and operating boundaries

Figure 3 shows held-out RVC by budget and the headroom–gain relation (full sweep inAppendix O), the ESS10 gain persists at 2048; Perspectrum and saturated ArgKP show distinct patterns. Across benchmarks the realized gains track available headroom (Pearson r=0.574, n=4, descriptive) and vanish where the ceiling is met—the suite maps the method's operating region, not a universal win.

Table 2: Sensitivity across encoder-specific pipelines at budget 1024 in the same scopes as Table 1. Each encoder column rebuilds embeddings, pools, and features together (not an encoder-only controlled comparison); the bge columns retain the earlier local-selection protocol (smaller pools; Appendix A). Left: exact-oracle headroom; right: paired LMVU–MMR ΔRVC with 95% CI (WVS MiniLM-L6, range across the three issue-group deltas). ArgKP is omitted because MMR and LMVU both reach the oracle ceiling under every encoder, giving zero headroom and zero ΔRVC.

	headroom (oracle − MMR)			LMVU − MMR ΔRVC [95% CI]		
	MiniLM-L6	bge-base	bge-m3	MiniLM-L6	bge-base	bge-m3
WVS	0.0833	0.0417	0.0000	+0.0833 [+0.000, +0.250]	+0.0417 [+0.000, +0.125]	−0.0417 [−0.125, +0.000]
Perspectrum	0.0691	0.0797	0.0730	+0.0161 [−0.001, +0.035]	**+0.0251** [**+0.004**, **+0.049**]	+0.0175 [−0.010, +0.048]
ESS10	0.1573	0.2065	0.1241	+0.0466 [−0.022, +0.118]	**+0.0741** [**+0.001**, **+0.145**]	+0.0286 [−0.028, +0.085]

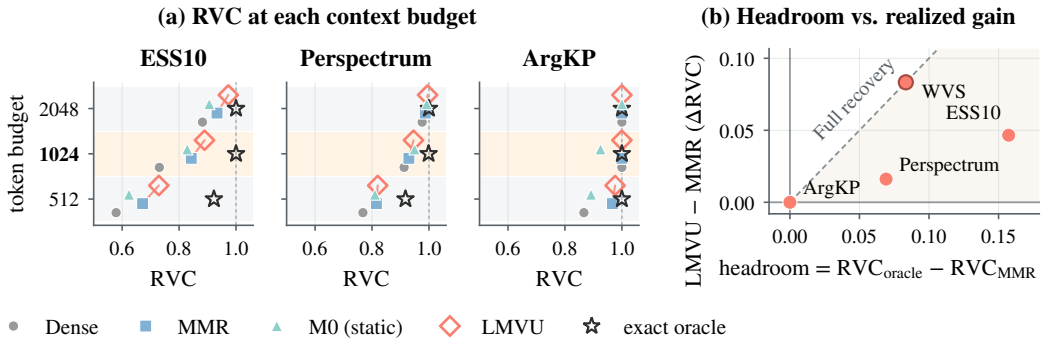


Figure 3: (a) Held-out RVC by budget (stars: exact oracle); ArgKP saturates at 1024/2048; LMVU reaches the WVS oracle at all budgets. (b) Headroom–gain at 1024 (Pearson r=0.574, n=4).

Budget scaling and gain concentration. The WVS gain over MMR is largest at 512 tokens (+0.117), persists at 1024 (+0.083), and narrows at 2048 (+0.042) as the relaxed budget lets MMR buy the missing viewpoints; ESS10 decays similarly (+0.057/ + 0.047/ + 0.040) but stays positive, while ArgKP's 512-token edge (+0.010) vanishes at saturation and Perspectrum's stays small (at most +0.016). At 1024 the WVS gain concentrates in the corruption-perception issue group (+0.250), whose evidence MMR's redundancy penalty crowds out; the remaining groups are solved by both allocators, leaving corruption as the sole margin.

6 Discussion and Conclusion

LMVU learns state-dependent evidence selection under a context budget, training on small development pools and selecting from full query-specific pools. Across four benchmarks, it improves viewpoint coverage over MMR on three, reaches the exact coverage ceiling on WVS, and matches saturated ArgKP. State conditioning benefits WVS and ESS10, while a candidate-only scorer remains stronger on Perspectrum. Gains over MMR also extend to repeated evidence support. For multi-viewpoint RAG queries, this allocation determines which supported answers and positions are available to the generator within the context budget.

Limitations. Performance depends on the embedding pipeline and viewpoint annotations for training and evaluation. The full-pool extension is exploratory (Appendix A), and the learned scorer has no coverage approximation guarantee. Further work should evaluate downstream generation and adaptation to domains requiring new viewpoint inventories.

AI USE STATEMENT

We used AI tools to aid and polish manuscript writing, including improving clarity and readability, and for retrieval and discovery of relevant literature to support the related-work discussion. The authors are responsible for verifying AI-assisted text and checking cited sources, and take full responsibility for the final manuscript, including its claims, results, and references.

REFERENCES

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. From arguments to key points: Towards automatic argument summarization. In *Proceedings of ACL*, pp. 4029–4039, 2020.

Jaime Carbonell and Jade Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of SIGIR*, 1998.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, 2024.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. Seeing things from a different angle: Discovering diverse perspectives about claims. In *Proceedings of NAACL*, 2019.

Esin Durmus, Karina Nguyen, Thomas I. Liao, Nicholas Schiefer, Amanda Askell, Anton Bakhtin, Carol Chen, Zac Hatfield-Dodds, Danny Hernandez, Nicholas Joseph, Liane Lovitt, Sam McCandlish, Orowa Sikder, Alex Tamkin, Janel Thamkul, Jared Kaplan, Jack Clark, and Deep Ganguli. Towards measuring the representation of subjective global opinions in language models, 2023.

Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

European Social Survey European Research Infrastructure (ESS ERIC). ESS10 integrated file, edition 3.2. Data set, Sikt – Norwegian Agency for Shared Services in Education and Research, 2023.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of EMNLP*, 2020.

Saadat Hasan Khan, Spencer Hong, Jingyu Wu, Kevin Lybarger, Youbing Yin, Erin Babinsky, and Daben Liu. DF-RAG: Query-aware diversity for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 2873–2894, 2026.

Sam Khuller, Anna Moss, and Joseph Naor. The budgeted maximum coverage problem. *Information Processing Letters*, 70(1):39–45, 1999.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. The prism alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models, 2024.

Alex Kulesza and Ben Taskar. *Determinantal Point Processes for Machine Learning*, volume 5 of *Foundations and Trends in Machine Learning*. 2012.

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

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2011.

Sewon Min, Kenton Lee, Ming-Wei Chang, Kristina Toutanova, and Hannaneh Hajishirzi. Joint passage ranking for diverse multi-answer retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6997–7008, 2021.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, Amin Karbasi, Jan Vondrák, and Andreas Krause. Lazier than lazy greedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14:265–294, 1978.

Chao Peng, Bin Wang, Zhilei Long, and Jinfang Sheng. AdaGreS: Adaptive greedy context selection via redundancy-aware scoring for token-budgeted RAG, 2025.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of EMNLP-IJCNLP*, pp. 3982–3992, 2019.

Rodrygo L. T. Santos, Craig Macdonald, and Iadh Ounis. Exploiting query reformulations for web search result diversification. In *Proceedings of the 19th International Conference on World Wide Web (WWW)*, pp. 881–890, 2010.

Taylor Sorensen, Liwei Jiang, Jena Hwang, Sydney Levine, Valentina Pyatkin, Peter West, Nouha Dziri, Ximing Lu, Kavel Rao, Chandra Bhagavatula, Maarten Sap, John Tasioulas, and Yejin Choi. Value kaleidoscope: Engaging ai with pluralistic human values, rights, and duties, 2023.

Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

Shuting Wang, Xin Yu, Mang Wang, Weipeng Chen, Yutao Zhu, and Zhicheng Dou. RichRAG: Crafting rich responses for multi-faceted queries in retrieval-augmented generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 11317–11333, 2025a.

Zhichao Wang, Bin Bi, Yanqi Luo, Sitaram Asur, and Claire Na Cheng. Diversity enhances an LLM's performance in RAG and long-context task, 2025b. URL `https://arxiv.org/abs/2502.09017`.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024.

Shijia Xu, Zhou Wu, Xiaolong Jia, Yu Wang, Kai Liu, and April Xiaowen Dong. Self-correcting RAG: Enhancing faithfulness via MMKP context selection and NLI-guided MCTS. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 20953–20976, 2026.

Wenlong Zhao, Debanjan Mondal, Niket Tandon, Danica Dillion, Kurt Gray, and Yuling Gu. Worldvaluesbench: A large-scale benchmark dataset for multi-cultural value awareness of language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pp. 17696–17706, 2024.

A Evaluation Protocol and Integrity

Development/confirmation discipline. All method decisions (feature schema, candidate pool, allocator, budgets, training configuration) were locked in development stages before any held-out access. The protected WVS fold (USA fold 4, n=6 queries) then saw a single-touch confirmation evaluation, followed only by the five predeclared baselines-only augmentations described below; no retuning occurred or was permitted at any point. Subsequently, the fixed-pool oracle was recomputed as an exact budgeted maximum-coverage optimum (§3); this required one additional read-only pass over the same gold annotations and changed no method, selection, tuning, or method-side number. The repeated-support audit of Appendix P followed the same pattern: it recomputed per-trajectory coverage under RVC_2 and the exact repeated-support oracle from the locked selection records, re-accessing the protected WVS gold once more, read-only, to recover the augmentation baselines'

selection sequences by deterministic replay. All twelve deterministic baselines reproduced their locked trajectories' RVC values bit-exactly; the random baseline was redrawn under a pinned seed with a predeclared sampling-noise gate (Appendix P). No method, pool, tuning, or previously reported number changed. Each transfer benchmark followed: adapter verification → fixed candidate pool → development training (grouped by query) → development method lock → pre-gold trajectory lock → single-touch held-out evaluation.

Label-blindness firewall. Runtime selection uses only the 18 runtime-visible features. For every benchmark, the runtime access audits recorded zero gold-field accesses and zero external provider calls. Gold annotations were used exclusively to construct development-side training targets, to guide the gold-visitation state-harvesting family on development queries (§4), and to compute the post-hoc coverage metrics used in the audit.

Determinism and seeds. The allocator is deterministic; across ten selection seeds $\{17, 29, 41, 53, 67, 79, 83, 97, 109, 127\}$ the action sequences are identical on all checked queries. Training uses a fixed seed (41). All similarity geometry uses the frozen sentence-transformers/all-MiniLM-L6-v2 encoder (snapshot `1110a243`, 384 dimensions, maximum sequence 256 tokens, normalized embeddings, local files only); the pipeline fails closed on any deviation from this identity. The full-pool extension fits three training seeds $\{17, 29, 41\}$ for the learned rows (M0, JPR, RichRAG, LMVU); given a fitted scorer, inference is deterministic and runs once per query, and the stochastic baselines (Random, xQuAD, Cluster) use the ten pinned seeds above.

Stopping rule. Dataset expansion was governed by a predeclared rule: stop when two WVS-external benchmarks reach positive status (met: Perspectrum, ESS10) or after at most three justified headroom-rich additions. The rule triggered on the first condition.

Classification tiers. Held-out outcomes use predeclared ordered tiers (strong positive / positive / boundary non-inferior / negative) defined by delta sign, magnitude, and headroom recovery; saturated non-inferiority does not count toward the positive target.

Baseline augmentation locks. The thirteen added baselines were predeclared in five baselines-only augmentation locks before their evaluation: RANDOM, BM25, FACLOC (lock v1); XQUAD, KDPP, SUBMIX, LEXRANK (lock v2); CERERANK, CLUSTER (lock v3); ADAGRES (Peng et al., 2025), MMKP (Xu et al., 2026) (lock v4); and JPR (Min et al., 2021), RICHRAG (Wang et al., 2025a) (lock v5). Lock v4 ran the published definitions on the same frozen MiniLM pool geometry as every other baseline: ADAGRES greedily adds the item maximizing query relevance minus an instance-adaptive redundancy penalty estimated from pool-level similarity means, stopping at a non-positive marginal gain; MMKP clusters each pool at a 0.82 cosine threshold (connected components), admits at most one passage per cluster, and solves the resulting budgeted relevance maximization exactly by Pareto-pruned dynamic programming; both are deterministic. Lock v5 adapts the two published neural rankers to state-aware selection on the same frozen pools and development states: JPR is a t5-base sequence-to-sequence reranker trained with a listwise softmax cross-entropy over the *yes* logit on inputs of the form query + current selection + candidate, and RICHRAG is a Qwen2.5-7B-Instruct LoRA adapter trained by supervised fine-tuning on yes/no new-perspective judgments followed by direct preference optimization on same-state winner/loser candidate pairs; at inference each greedily adds its highest-scoring candidate, scoring only the top-64 dense-relevant budget-feasible candidates per step, with per-benchmark fits (WVS per dependency fold) under the three training seeds and deterministic forward-pass inference. For CERERANK, the cross-encoder (bge-reranker-v2-m3, revision 953dc6f) scored every frozen pool item exactly once before selection; the scores were cached with their input hashes and shared across all seeds, making the baseline deterministic, and the model is self-hosted with zero external provider calls. Each lock fixed the baseline definitions, budgets $\{512, 1024, 2048\}$, and seeds $\{17, \ldots, 127\}$ before any run, with no change to the frozen LMVU method, pools, or previously reported results, and no retuning. The WVS protected fold was re-accessed for these fixed baselines only: once per lock for v1–v3, for lock v4 once under the original local-pool protocol and once under this version's full-pool protocol, both fixed by the same lock, and for lock v5 once under the full-pool protocol. LMVU's protected result was neither recomputed nor modified.

Full-pool extension (this version). This version changes where selection happens: training remains confined to the frozen local pools $P(q)$, while all reported evaluation runs select from the full prequalified evidence pools $E(q)$. The model class, objective, and training configuration are unchanged, but training states are re-harvested under the delivered pipeline (§4)—budget-indexed local pools and five development state-visitation families—and every learned row (WVS per dependency fold, each transfer benchmark, M0 under its three-feature mask) is refit on these states with three training seeds $\{17, 29, 41\}$. The extension is built on an independently delivered LF (local-training/full-selection) matrix whose bundle passed integrity verification (`verify_bundle`), whose $5{,}148$ learned (LMVU) trajectories were bit-reproduced by parameter replay (of $7{,}293$ delivered runs), and whose MMR and Dense rows were reconciled per query (tolerance 10^{-9}) against an independent offline rerun on all four benchmarks; that rerun reproduced all deterministic rows exactly and flipped only three near-tie LMVU rows, traced to cross-version refit noise, and none of this version's numbers is drawn from those rerun rows. The sixteen non-oracle baselines were then rerun on the full pools under their locked scorer definitions—the original twelve plus the lock-v4 ADAGRES and MMKP and the lock-v5 JPR and RICHRAG, ported onto the same pool matrix—gated on per-query reproduction ($\leq 10^{-9}$) of the delivered MMR and Dense rows; CERERANK reused the same self-hosted cross-encoder, and the stochastic baselines were redrawn under ten pinned seeds. The exact oracles (breadth and repeated support) were recomputed on the full pools; the repeated-support oracle was verified by an in-process identical second run, dominates every method trajectory on every query–budget cell, and reproduces the locked local-pool audit exactly when restricted to the original pools. Because this extension follows previously inspected held-out results, its evaluations are exploratory rather than a fresh confirmatory test. All development-fold diagnostics (Appendices F, G, I, H, M), the embedding sweep (Table 5) and the bge encoder columns of Table 2, and the Δ_2 ablation of Appendix P are artifacts of the original local-selection development pipeline and are reported unchanged. The case-study tables and figures in Appendix L instead display the evaluation trajectories obtained by selecting from the full pools in this version.

B EXPERIMENTAL SETUP

Datasets. Table 3 summarizes the four viewpoint-annotated benchmarks. WVS (Zhao et al., 2024) contributes survey topics with component-level atomic claims; we evaluate on a protected respondent fold under the access discipline of §A. Perspectrum (Chen et al., 2019) contributes claim–perspective units, ArgKP (Bar-Haim et al., 2020) argument key points (included as a saturation control), and ESS10 (European Social Survey European Research Infrastructure (ESS ERIC), 2023) European Social Survey round-10 items. Development and held-out splits are grouped by query so that no query appears in both.

Table 3: Benchmark summary. Valid viewpoints/query and full selection-pool rows/query are means (the full pools are budget-independent); training uses the smaller MMR-oversampled pools below. WVS dev and evaluation use the same six queries (3 topics $\times$ 2 paraphrases) but disjoint gold matrices; evaluation uses the protected respondent fold. Cell shading scales by value per column.

Benchmark	Dev n	Eval n	Valid viewpoints/query	Pool rows/query
WVS (protected)	6	6	6.0	507.3
Perspectrum	32	97	3.0	8.9
ArgKP	32	6	5.5	75.7
ESS10	152	34	4.8	88.3

Pool construction (training and selection). All methods share identical pools. The *selection* pool $E(q)$ is each query's full prequalified evidence set: deduplicated, budget-independent, with token costs from the adapter tokenizer's whole-evidence token lengths; the runtime candidate policy enumerates exactly the budget-feasible pool members. The smaller *training* pool $P(q)$ is the frozen MMR-oversampled candidate set of the original pipeline (maximal marginal relevance with λ=0.5 over dense query–candidate scores and embedding novelty, filled to $4\times$ the budget in tokens) and is used only to harvest development training states (§4). The pools are deliberately redundant (multiple

rows per viewpoint; Table 3). Coverage therefore depends on which evidence the allocator selects, rather than being fixed by a minimally populated pool.

Training. Training targets are realized marginal coverage gains $\Delta V(e \mid q, S)$ on development states only. For each state we form within-state candidate pairs (e^+, e^-) with $\Delta V(e^+) > \Delta V(e^-)$, capped at 20 pairs per state (deterministic hash-seeded subsample), and fit a logistic pairwise ranker on standardized symmetric feature differences (C=1.0, LBFGS, max. 300 iterations, seed 41). WVS fits one model per query-dependency fold and each transfer benchmark fits per-benchmark coefficients under the same configuration; every learned row reported in this version is refit on the harvested development states of §4 with three training seeds $\{17, 29, 41\}$. The hyperparameter setting is shared and predeclared across benchmarks, and no fitting uses held-out data.

C Baseline Definitions

All baselines share the full selection pools and token accounting of this version's full-pool rerun (Appendix A); they differ only in the per-step scoring rule. RANDOM: uniform random priority order per query (10 seeds, mean reported). BM25: hand-rolled BM25 (k_1=1.2, b=0.75) over the query's pool passagess, score static across steps. FACLOC: facility-location greedy over frozen sentence embeddings; the per-step score of candidate e is $\sum_{j \in E(q)} \max\left(0,\ \mathrm{sim}(e, j) - c_j\right)$ with c_j the running best coverage of pool item j, a relevance-free monotone submodular coverage objective.

Four citation-backed baselines (lock v2) are local approximations over the same full pools. XQUAD (Santos et al., 2010): explicit aspect diversification with pseudo-aspects from spherical k-means over pool embeddings ($k = \min(10, \max(2, \lceil\sqrt{|E|}\rceil))$, label-blind), λ=0.5, hard aspect membership. KDPP (Kulesza & Taskar, 2012): greedy MAP inference over an L-ensemble with quality q_i the min-max-normalized dense relevance (shifted positive) and kernel S the clipped cosine similarity; incremental Cholesky updates. SUBMIX (Lin & Bilmes, 2011): α=0.5 mixture of normalized dense relevance and the normalized facility-location gain. One disclosure: with the gain divided by pool size, the coverage term's spread across candidates is $\sim 10^{-3}$ against $\sim 10^{-1}$ for the min-max relevance term on these pools, so the mixture degenerates to dense top-k on WVS and on all but one ESS10 query at the primary budget (per-query RVC identical to full precision); we report the baseline exactly as predeclared in the lock rather than renormalizing post hoc. LEXRANK (Erkan & Radev, 2004): PageRank centrality (damping 0.85, 50 iterations) over the cosine-similarity graph of the pool thresholded at 0.1; budget-feasible top-k.

Two modern baselines (lock v3) follow the same full-pool discipline. CERERANK: every pool item is scored once by the cross-encoder reranker bge-reranker-v2-m3 (Chen et al., 2024) (query–evidence text pairs truncated at 512 tokens, fp16 inference, scores cached with input hashes and shared across all ten seeds), and selection is budget-feasible greedy on the cached score. CLUSTER: spherical k-means over the frozen pool embeddings ($k = \min(12, \max(2, \lceil\sqrt{|E|}\rceil))$, seeded initialization, 25 iterations); clusters are ordered by the dense relevance of their most relevant member, and selection rotates round-robin across clusters, taking each cluster's next most relevant item, yielding a static rank order that is consumed budget-feasibly.

Two allocation heuristics (lock v4) and two training-heavy neural rankers (lock v5) complete the panel; Appendix A gives their locked configurations. ADAGRES (Peng et al., 2025): greedy query relevance minus an instance-adaptive pairwise-similarity redundancy penalty, stopping at a non-positive marginal gain. MMKP (Xu et al., 2026): at most one item per 0.82-cosine connected-component cluster, with exact Pareto-pruned dynamic programming for the budgeted relevance maximization. JPR (Min et al., 2021): t5-base sequence-to-sequence reranker trained with a listwise softmax over the *yes* logit on query + current selection + candidate inputs. RICHRAG (Wang et al., 2025a): Qwen2.5-7B LoRA adapter trained by SFT on yes/no new-perspective judgments followed by DPO on same-state candidate pairs; both neural rankers score only the top-64 dense-relevant budget-feasible candidates per step.

D PAIRED STATISTICS PROTOCOL

For each benchmark at budget 1024 we pair per-query RVC (seed-averaged where applicable: ten pinned seeds for the stochastic baselines, three training seeds for the learned rows) between LMVU and each comparator, then compute the mean delta, a paired bootstrap 95% CI (10,000 replicates, fixed seed), win/tie/loss counts at tolerance 10^{-9}, and the two-sided sign-test p-value. On WVS and ArgKP, whose six queries form only three independent issue/topic groups, no intervals are reported and statistics are descriptive; on Perspectrum and ESS10 a Wilcoxon signed-rank check accompanies the sign test. All quantities are computed from locked per-query metric artifacts without re-evaluation. Table 12 reports the paired statistics for LMVU − MMR as the reference comparison (breadth at the primary budget; depth across budgets). Most queries tie near the coverage ceiling, so we emphasize effect sizes and uncertainty intervals rather than drawing separate significance claims for each benchmark.

E FORMAL ANALYSIS

This section collects the formal statements behind the assertions of §3 and §4. We reuse the notation established there: a query q carries a viewpoint inventory V_q; each passage e covers $V(e) \subseteq V_q$ at token cost $c(e)$; the coverage value of a selection is $F_q(S) = |V(S)|$ with $V(S) = \bigcup_{e \in S} V(e)$, so that $\mathrm{RVC}(S) = F_q(S)/|V_q|$; the budget is B; and the realized marginal gain of a candidate in state (q, S) is $\Delta V(e \mid q, S) = F_q(S \cup \{e\}) - F_q(S)$.

The coverage objective.

Proposition 1 (Monotonicity and submodularity)**.** *For every query q, the coverage function F_q is monotone and submodular over all subsets of the full evidence pool $E(q)$.*

Proof. Monotonicity: $A \subseteq B$ implies $V(A) \subseteq V(B)$, hence $F_q(A) \leq F_q(B)$. Diminishing returns: for $A \subseteq B \subseteq E(q)$ and $e \notin B$,

$$\Delta V(e \mid q, A) \;=\; |V(e) \setminus V(A)| \;\geq\; |V(e) \setminus V(B)| \;=\; \Delta V(e \mid q, B),$$

where $V(A) \subseteq V(B)$ ensures that enlarging the selection cannot increase the marginal gain. □

The full-pool oracle is exact.

Proposition 2 (Exactness and cost of the bitmask oracle)**.** *Let $m_q = |V_q|$ and fix an enumeration of V_q, so each $V(e)$ is an m_q-bit mask. For every mask μ, the subset dynamic program stores the minimum total token cost of a sub-selection whose covered mask equals μ. After all pool items have been processed, the best budget-feasible entry in this table gives the exact value $\max_{S \subseteq E(q),\, c(S) \leq B} F_q(S)$. The computation uses $O\big(|E(q)| \cdot 2^{m_q}\big)$ transitions to process the items and $O(2^{m_q})$ memory for the table of minimum costs.*

Proof. Process the items $e_1, \ldots, e_n$ of $E(q)$ in any order and maintain $\mathrm{dp}_i[\mu]$ = the minimum cost of a subset of $\{e_1, \ldots, e_i\}$ covering exactly mask μ ($+\infty$ if unattainable). The transitions $\mathrm{dp}_{i+1}[\mu] \leftarrow \min\big(\mathrm{dp}_{i+1}[\mu], \mathrm{dp}_i[\mu]\big)$ (skip) and $\mathrm{dp}_{i+1}[\mu \text{ OR } V(e_{i+1})] \leftarrow \min\big(\mathrm{dp}_{i+1}[\mu \text{ OR } V(e_{i+1})], \mathrm{dp}_i[\mu] + c(e_{i+1})\big)$ (take) preserve the invariant: by induction, every subset of the first $i{+}1$ items either excludes e_{i+1} or includes it, and each case is costed optimally by the corresponding transition. Since F_q depends on S only through its covered mask, the optimum is $\max\{\, |\mu| : \mathrm{dp}_n[\mu] \leq B \,\}$. There are 2^{m_q} masks and each item triggers one transition per mask. □

On all four benchmarks $m_q \leq 9$ (viewpoint inventories, unchanged by the pool), so a query's table has at most $2^9 = 512$ mask states: the exact per-query solve on the full pool is negligible, which is why the paper reports exact rather than approximate oracle values.

The repeated-support oracle is exact.

Proposition 3 (Exactness and cost of the trit-state oracle)**.** *Let $m_q = |V_q|$ and fix an enumeration of V_q and a source map* src*. Assume that within the query, distinct pool items carry distinct source units—verified on all four benchmarks, where the source unit is the respondent (WVS, ESS10), the document itself (Perspectrum), or the item itself (ArgKP). The dynamic program that stores, for every vector $\tau \in \{0,1,2\}^{m_q}$ of capped per-viewpoint source counts, the minimum total token cost of a sub-selection realizing exactly τ returns $\max_{S\subseteq E(q),\, c(S)\le B} \sum_v \min\{n_v(S), 2\}$ exactly, using $O\big(|E(q)| \cdot 3^{m_q}\big)$ transitions and $O(3^{m_q})$ memory. The pool may first be preprocessed by keeping only a cheapest item for each pair (π, s) of support pattern $\pi \subseteq V_q$ and source s. This reduction preserves the optimal value of the repeated-support objective.*

Proof. Encode τ as a base-3 integer and maintain $\mathrm{dp}_i[\tau] =$ the minimum cost of a subset of the first i items whose capped source-count vector is exactly τ ($+\infty$ if unattainable). Taking item e_{i+1} with pattern π and source s maps τ to τ' with $\tau'_v = \min\{\tau_v + 1, 2\}$ for $v \in \pi$ and $\tau'_v = \tau_v$ otherwise. Under the injectivity hypothesis, each taken item introduces a source unseen by any other taken item, so this per-item increment is exactly the increment of capped *distinct-source* counts. The skip/take transitions preserve the min-cost invariant by the same induction as Proposition 2, and the objective depends on S only through τ, so the optimum is $\max\{\sum_v \tau_v : \mathrm{dp}_n[\tau] \le B\}$. For the preprocessing: two items sharing both pattern π and source s contribute the identical increment, so any selection containing both can replace the costlier one by the cheaper one (or drop it) without changing the realized τ; an optimum therefore survives among the retained items. . □

With $m_q \le 9$ the table holds at most $3^9 = 19{,}683$ states. Exact repeated-support optimization therefore remains inexpensive for the viewpoint inventories evaluated here.

Why raw-gain greedy has no knapsack guarantee.

Proposition 4 (Unbounded approximation ratio under a knapsack budget)**.** *For every $\varepsilon > 0$, there is an instance of the budgeted viewpoint-coverage problem on which budget-feasible raw-gain greedy returns a selection S^{gr} with $F_q(S^{\mathrm{gr}})/F_q(S^\star_q) < \varepsilon$. Here raw-gain greedy denotes the loop of Algorithm 1 scored by the true gains ΔV.*

Proof. Fix an integer $m \ge 3$ and let $V_q = \{v_1, \dots, v_{m+1}\}$ and $B = m$. The pool contains an item a with $c(a) = m$ and $V(a) = \{v_1, v_2\}$, and items $b_1, \dots, b_{m+1}$ with $c(b_i) = 1$ and $V(b_i) = \{v_i\}$. Raw-gain greedy first compares $\Delta V(a \mid q, \emptyset) = 2$ against $\Delta V(b_i \mid q, \emptyset) = 1$ and selects a, which exhausts the budget, so $F_q(S^{\mathrm{gr}}) = 2$. The optimum selects any m of the b_i, giving $F_q(S^\star_q) = m$. The ratio $2/m$ falls below ε for all $m \ge \lceil 2/\varepsilon \rceil$. □

Remark 5. *Under equal unit costs (a cardinality constraint) the same raw-gain greedy rule recovers the classical $(1 - 1/e)$ guarantee (Nemhauser et al., 1978), and under a knapsack budget $(1 - 1/e)$ is restored by density-aware partial-enumeration variants (Khuller et al., 1999; Sviridenko, 2004). Every such rule conditions each step on the true marginal gains ΔV—exactly what a label-blind runtime allocator cannot observe (§3). This is the formal motivation for learning a surrogate gain ordering, and the reason the learned plug-in claims no constant-factor guarantee (§4).*

Surrogate fidelity implies coverage. To isolate the role of the learned ordering we analyze an idealized unit-cost case in which every item costs one token, so the budget is the cardinality constraint $k = B$ and the allocator takes exactly k steps.

Proposition 6 (Per-step shortfall bound)**.** *Suppose unit costs, and let the surrogate-greedy trajectory $S_0 \subsetneq S_1 \subsetneq \cdots \subsetneq S_k$ satisfy, at every step t, the per-step shortfall condition*

$$\Delta V(e_t \mid q, S_{t-1}) \;\ge\; \max_{e \in E(q)\setminus S_{t-1}} \Delta V(e \mid q, S_{t-1}) \;-\; \eta_t.$$

Then

$$F_q(S_k) \;\ge\; \Big(1 - \tfrac{1}{e}\Big)\,\mathrm{OPT}_k \;-\; \sum_{t=1}^{k} \eta_t, \qquad \mathrm{OPT}_k \;=\; \max_{|S|\le k} F_q(S).$$

Proof. Write $F_t = F_q(S_t)$ and let O be an optimal size-k selection. At step t,

$$F_t - F_{t-1} \;\geq\; \max_e \Delta V(e \mid q, S_{t-1}) - \eta_t \;\geq\; \tfrac{1}{k}\textstyle\sum_{o\in O} \Delta V(o \mid q, S_{t-1}) - \eta_t$$
$$\geq\; \tfrac{1}{k}\big(F_q(S_{t-1}\cup O) - F_{t-1}\big) - \eta_t \;\geq\; \tfrac{1}{k}\big(\mathrm{OPT}_k - F_{t-1}\big) - \eta_t,$$

where the second inequality averages the gains over O, the third telescopes them by submodularity, and the fourth uses monotonicity. With $\delta_t = \mathrm{OPT}_k - F_t$ this reads $\delta_t \leq \big(1-\frac{1}{k}\big)\,\delta_{t-1} + \eta_t$; unrolling from $\delta_0 \leq \mathrm{OPT}_k$ and using $\big(1-\frac{1}{k}\big)^k \leq 1/e$ (and dropping the geometric weights on the η_t, each at most one) gives $\delta_k \leq \mathrm{OPT}_k/e + \sum_{t\leq k}\eta_t$. □

Remark 7. *Three connections back to the paper. (i) The shortfall η_t is the quantity that the within-state ranking objective of §4.2 optimizes a convex surrogate of: a state contributes zero shortfall whenever the model's top-ranked candidate attains the state's maximum realized gain, so the development-fold ranking and learnability metrics (Appendices F and G) are empirical proxies for $\sum_t \eta_t$. (ii) When the within-state orderings are perfect, the surrogate trajectory coincides step-for-step with the trajectory obtained by greedily selecting on true gains. (iii) The cardinality induction requires equal costs and does not apply to the unequal token costs in our experiments. This limitation is consistent with Proposition 4 and the absence of a constant-factor claim in §4.*

MMR is a fixed-coefficient special case.

Remark 8 (MMR as a static restriction of the hypothesis class)**.** *The MMR per-step score $\lambda\,\mathrm{rel}(q,e) - (1-\lambda)\max_{e'\in S}\cos(e,e')$ is a two-term linear form in LMVU's feature vector $\phi(q,S,e)$ (feature groups one and two of Table 8) with coefficients frozen at $(\lambda, -(1-\lambda))$—up to per-dataset standardization and positive rescaling, neither of which changes the within-state argmax. The learned hypothesis class of §4.2 therefore contains the MMR family; learning θ replaces the single fixed relevance–novelty trade-off with a state-dependent one, formalizing the remark in §4.2.*

F Development-Fold Evidence

Development-fold cross-validation (grouped by query; never used for held-out claims) at budget 1024 (cell shading scales with RVC within each column):

Benchmark (dev n)	Dense	MMR	M0	M1 (LMVU)
Perspectrum (32)	0.9182	0.8964	**0.9714**	0.9583
ArgKP (32)	0.9771	0.9885	0.9797	**0.9896**
ESS10 (152)	0.7318	0.8079	0.8381	**0.8408**

The development pattern matches the held-out one: M1 improves over MMR on Perspectrum ($+0.062$) and ESS10 ($+0.033$), the M0-over-M1 inversion on Perspectrum is already visible in development, and gains are small on near-saturated ArgKP.

G Pre-Registered Learnability Probe

Before any sequential evaluation we tested, on development states only, whether marginal viewpoint utility is learnable at all. The probe paired 117,387 runtime-feature rows (7,122 states, 70 development queries across WVS, Perspectrum, and ArgKP) with hidden marginal-gain targets, and compared frozen heuristic rankers and the learned models under grouped cross-validation (identical states and candidate sets for every method). Table 4 reports the within-state ranking quality measured by the probe. The pre-registered gate required M1 to outperform M0 on NDCG@5, state oracle regret, and top-1 positive-gain rate; the probe met all three criteria.

Two observations motivate the method design. First, no runtime-visible heuristic approaches the oracle, and dense relevance is *worse than random* on within-state ranking—relevance does not order marginal gains. Second, the state-dependent model (M1) is the only method that beats every heuristic on all three criteria, which is what licensed sequential training.

Table 4: Learnability probe on development states pooled over WVS, Perspectrum, and ArgKP: within-state NDCG and mean state oracle regret (lower is better). The heuristic rankers are frozen, while M0 and M1 are learned. The oracle ranks candidates by their true marginal gain. Shading is column-wise over the seven non-oracle rows, with regret reversed so that darker means lower regret. The oracle row has a uniform gray background and is excluded from each column's color scale.

Method	NDCG@5	NDCG@10	Oracle regret
Random	0.5554	0.6165	0.5638
Dense relevance	0.5243	0.5837	0.5804
MMR-style	0.6313	0.6925	0.4789
Soft novelty	0.6223	0.6801	0.4706
Static validity	0.5707	0.6308	0.5393
M0 (static learned)	0.5869	0.6513	0.5525
M1 (LMVU)	**0.6575**	**0.7165**	**0.4516**
Oracle marginal gain	1.0000	1.0000	0.0000

H Feature Ablations

Static vs. state-dependent. M0 (three static features) vs. M1 (full state-dependent features) within-state ranking on development states (cell shading scales with value within each column; regret columns flipped, darker is lower):

	NDCG@5		Oracle regret	
Dataset	M0	M1	M0	M1
WVS	0.3961	**0.6597**	1.3945	**0.8554**
Perspectrum	0.8393	**0.8413**	**0.2537**	0.2688
ArgKP	0.3704	**0.4732**	0.6934	**0.5587**

State-dependence drives the ranking-quality gain on WVS and ArgKP; on Perspectrum the two are near-identical, foreshadowing the held-out M0>M1 inversion discussed in §6.

Atomic features. Removing the six atomic-granularity features from M1 (12-feature variant, retrained on the same development states) leaves Perspectrum and ArgKP unchanged (NDCG@5 deltas ≤ 0.0002) but sharply degrades WVS: NDCG@5 $0.6597 \rightarrow 0.4418$, top-1 positive-gain rate $0.5425 \rightarrow 0.4088$. This WVS result highlights the value of retaining atomic structure when individual survey statements express more than one distinct claim.

I Parameter Sensitivity (Development Only)

Figure 4 shows dev-CV RVC at budget 1024 when varying the two training-side hyperparameters around the frozen operating point (20 pairs/state, C=1.0; reproduced exactly: Perspectrum 0.9583, ESS10 0.8408). ESS10 is flat across both axes (RVC within $[0.838, 0.842]$); Perspectrum varies mildly with the training-pair cap (0.9406 at 5 pairs, 0.9625 at 10, 0.9583 at 20) and is insensitive to C over two orders of magnitude (0.9583 at $C \in \{0.1, 1, 10\}$). The method does not rely on a finely tuned operating point. These runs accessed development folds only; no held-out or protected data were accessed during the sensitivity analysis.

J Embedding-Model Robustness

The canonical pipeline freezes one encoder (sentence-transformers/all-MiniLM-L6-v2; §5). We predeclared a sweep over bge-base-en-v1.5 (Xiao et al., 2024) and bge-m3 (Chen et al., 2024), rerunning embeddings, pools, baselines, training, and the exact oracle at budgets $\{512, 1024, 2048\}$ with the locked configuration. A reimplementation gate reproduced canonical pools and MMR/oracle

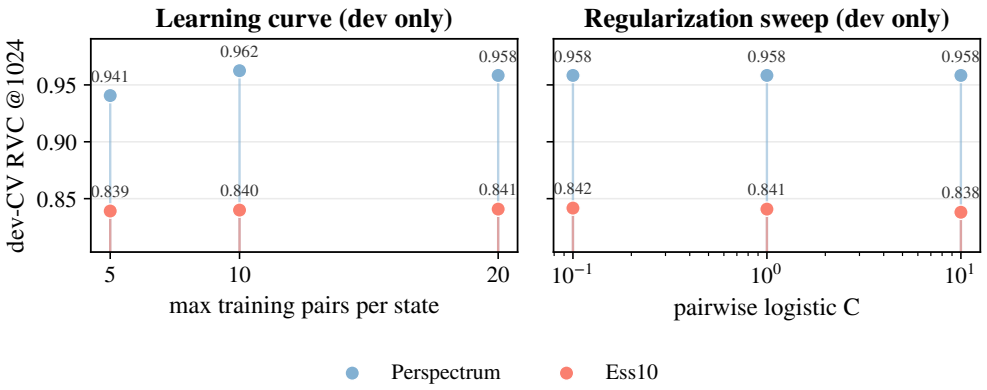


Figure 4: Training-side sensitivity on development folds (grouped CV, budget 1024). Left: learning curve over the per-state pair cap. Right: regularization sweep.

Table 5: Embedding-model robustness at budget 1024 in the main-text scopes (WVS: protected fold; Perspectrum, ArgKP, ESS10: held out): exact-oracle headroom (oracle − MMR), RVC of M1 and MMR, and the paired M1 − MMR delta with 95% bootstrap CI and win/tie/loss counts. All rows are retained from the original local-selection protocol (smaller candidate pools; Appendix A). A panel-level summary appears as Table 2; WVS development-fold metrics remain in the sweep artifact. Cell shading scales with value within each column (headroom by magnitude; M1/MMR darker is higher; the Δ column diverging: red positive, blue negative, zero unshaded).

Benchmark	Encoder	Headroom	M1	MMR	Δ [95% CI]	W/T/L
WVS	MiniLM-L6 (canonical)	0.0417	0.9583	0.9167	+0.0417 [+0.0000, +0.1250]	1/5/0
	bge-base-en-v1.5	0.0417	0.9167	0.8750	+0.0417 [+0.0000, +0.1250]	1/5/0
	bge-m3	0.0000	0.9167	0.9583	−0.0417 [−0.1250, +0.0000]	0/5/1
Perspectrum	MiniLM-L6 (canonical)	0.0663	0.9392	0.9296	+0.0096 [−0.0069, +0.0283]	7/85/5
	bge-base-en-v1.5	0.0797	0.9424	0.9173	+0.0251 [+0.0042, +0.0491]	10/84/3
	bge-m3	0.0730	0.9420	0.9246	+0.0175 [−0.0100, +0.0475]	12/78/7
ArgKP	MiniLM-L6 (canonical)	0.0000	1.0000	1.0000	+0.0000 [+0.0000, +0.0000]	0/6/0
	bge-base-en-v1.5	0.0000	1.0000	1.0000	+0.0000 [+0.0000, +0.0000]	0/6/0
	bge-m3	0.0000	1.0000	1.0000	+0.0000 [+0.0000, +0.0000]	0/6/0
ESS10	MiniLM-L6 (canonical)	0.1465	0.8866	0.8427	+0.0438 [−0.0220, +0.1142]	9/19/6
	bge-base-en-v1.5	0.2065	0.8617	0.7876	+0.0741 [+0.0010, +0.1452]	14/15/5
	bge-m3	0.1241	0.9046	0.8759	+0.0286 [−0.0277, +0.0853]	9/19/6

macros to 10^{-9}. WVS's protected-fold row used a predeclared evaluation-only amendment with no fitting or tuning. Table 2 summarizes the primary-budget comparison; Table 5 gives full results.

Two readings, both conditional on the representation. First, headroom remains necessary in every cell: across the 24 encoder×benchmark×budget cells of the two reported encoders, the seven with zero headroom (all ArgKP cells except bge-base at 512; bge-base WVS at 2048; bge-m3 WVS at 1024) never admit a positive delta—where MMR is already oracle-optimal there is nothing to recover. Second, headroom is not sufficient: of the 17 positive-headroom cells the delta is positive in 14 and negative in three, and all three negative cells arise under bge-m3 (Perspectrum at 512 and 2048; ESS10 at 2048) with intervals that include zero. On the held-out positive-headroom cells the delta is positive in 10 of 13, with four intervals excluding zero on the positive side and none on the negative side—all four under bge-base-en-v1.5 (Perspectrum at 512 and 1024; ESS10 at 512 and 1024), which also recovers the small ArgKP headroom at 512 in full and reproduces the original-protocol $+0.042$ on the WVS protected fold at the primary budget. Under bge-m3 the pattern is directionally preserved but attenuated: positive deltas at the primary budget on Perspectrum and ESS10 with intervals including zero, and one negative protected-fold cell (WVS at 1024, -0.042,

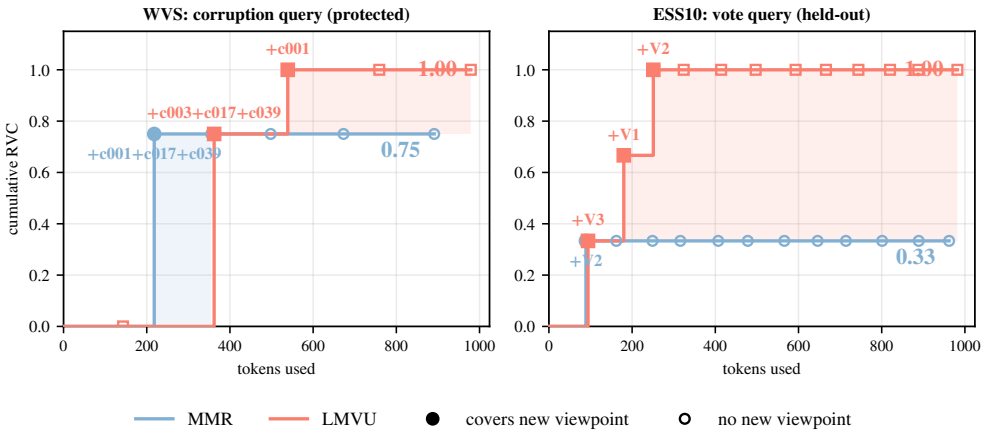


Figure 5: Cumulative RVC vs. tokens used for the two case-study queries at budget 1024. Filled markers are picks that newly cover at least one viewpoint; open markers are redundant picks; each curve's final RVC is shown in bold.

interval including zero; its 512 and 2048 cells are $+0.042$). The original-protocol result replicates under the encoder closest in family and scale but attenuates under the more distant representation; the frozen encoder is part of the method's operating point, and we claim no representation-free gains.

K Failure-Case Tally

From the development-state diagnostic (same states as Appendix G), we tally states where M1's chosen action underperforms. *M1 beats MMR*: 195 (WVS), 67 (Perspectrum), 432 (ArgKP) states. *M1 underperforms MMR*: 34, 65, 595. *M1 takes a zero-gain action while a positive-gain action exists*: 200, 336, 2113—concentrated in mid/late allocation steps (pooled: 528 early / 1157 mid / 964 late), where most remaining candidates add no new viewpoint. Failure mass is thus largest on ArgKP, the benchmark where everything saturates and the ordering error is inconsequential, and smallest relative to opportunity on WVS, where the held-out gain is largest.

L Case Studies

This section displays trajectories from the full-pool evaluation runs of this version (Appendix A); gold viewpoint annotations are used to label picks only. Figure 5 gives the cumulative-coverage view; the tables below list the picks themselves.

L.1 WVS protected fold: corruption query

The corruption query asks how respondents differ in their assessments of corruption across institutions; the annotated inventory has four viewpoints (pervasive institutional corruption, institutions perceived as corruption-free, and two low-perceived-corruption facets). Table 6 lists both allocations. MMR opens with a long, high-relevance item (218 tokens) that covers three viewpoints at once, then spends its remaining four picks on items whose viewpoints are already covered; the corruption-free viewpoint (c003) is never selected, leaving RVC at 0.750 with 891 tokens used. LMVU opens with a short item supporting no annotated viewpoint, then at step 1 selects a 219-token response covering the missing c003 together with c017/c039, and at step 2 adds c001, reaching RVC 1.000 within 539 tokens; its two remaining picks are redundant, for 979 tokens in total. The divergence is not a relevance decision—MMR's first pick is a good one—it is MMR's inability to discount redundancy against the as-yet-uncovered viewpoint.

Figure 6 shows the geometry behind Table 6. Items supporting the pervasive-corruption viewpoint c001 dominate the 508-item pool (223 items, 89 of them supporting only c001), while the

Table 6: Pick sequences for the WVS corruption query at budget 1024 (seed 17; the three training seeds coincide on this query): (a) MMR, (b) LMVU. Pick: anonymized respondent number (1–10 corruption-perception response). Tokens: per-pick cost. New: newly covered viewpoints (gold, display only). RVC: cumulative. Consecutive picks that newly cover nothing are grouped into one row (step range, pick count, tokens summed). Tags: c001 = pervasive institutional corruption, c003 = institutions perceived as corruption-free, c017/c039 = low perceived institutional corruption. RVC column shaded by magnitude (darker = higher coverage).

(a) MMR

Step	Pick	Tokens	New	RVC
0	0260 (6/10)	218	c001+c017+c039	0.750
1–4	4 picks	673	—	0.750

(b) LMVU

Step	Pick	Tokens	New	RVC
0	0008 (5/10)	143	—	0.000
1	0238 (10/10)	219	c003+c017+c039	0.750
2	0280 (10/10)	177	c001	1.000
3–4	2 picks	440	—	1.000

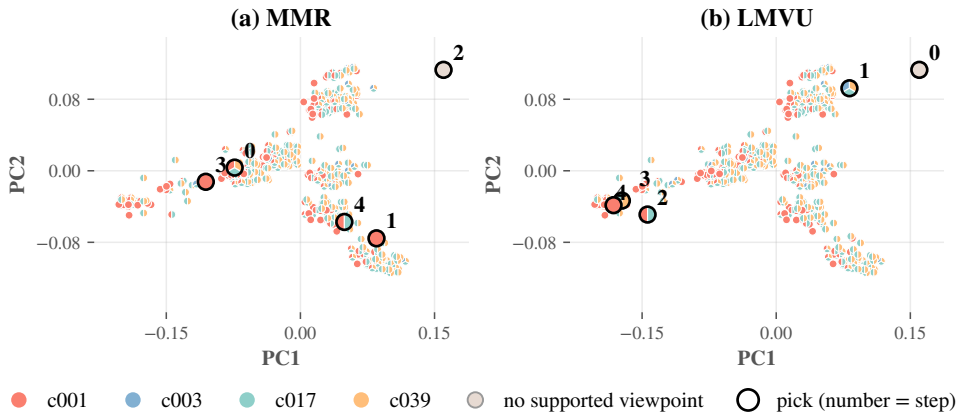


Figure 6: Full selection pool of the WVS corruption query (508 items, budget 1024) in its first two principal components. Colors show gold-supported viewpoints (pie markers: several viewpoints; gold for display only); numbered rings give pick order (Table 6). (a) MMR covers c001/c017/c039 with its first pick, and three of its four remaining picks are further c001 carriers that add no new viewpoint; (b) LMVU's picks span the pool—from the no-viewpoint item both methods select at one end to one of the sixteen c003 carriers at step 1—and reach full coverage at step 2.

corruption-free viewpoint c003 is carried by sixteen items. MMR covers c001/c017/c039 with its first pick, but three of its four remaining picks are further c001 carriers, so 673 of its 891 tokens revisit already-covered viewpoints; LMVU's picks fan out across the pool, and its step-1 pick is one of the sixteen c003 carriers. Both allocators also select the same query-relevant item that supports no annotated viewpoint (respondent 0008), a relevance-driven choice the two methods share.

L.2 ESS10 HELD-OUT: NON-VOTING QUERY

The ESS10 query asks whether the respondent voted in the last national election; the three annotated viewpoints are V1 = "No", V2 = "Yes", V3 = "Not eligible to vote". In Table 7, all 12 MMR picks support the same answer (RVC 0.333, 962 tokens), while LMVU covers all three viewpoints in its first three picks—V3 ("Not eligible to vote") at step 0, V1 ("No") at step 1, V2 ("Yes") at step 2—then makes nine redundant picks (RVC 1.000, 982 tokens).

L.3 A DEVELOPMENT FAILURE STATE

For balance, one development-only state (WVS political-participation query, step 0, dense trajectory family) where M1 underperforms, the MMR-style choice (respondent 1017) realizes a marginal gain

Table 7: Pick sequences for the ESS10 held-out vote query at budget 1024 (seed 17; the coverage-relevant first three picks are identical across the three training seeds): (a) MMR, (b) LMVU. Pick: anonymized respondent (country, gender, age group); remaining columns as in Table 6; consecutive picks that newly cover nothing are grouped into one row (step range, pick count, tokens summed). Tags: V1 = "No", V2 = "Yes", V3 = "Not eligible to vote". RVC-column shading encodes magnitude; darker cells indicate higher cumulative viewpoint coverage.

(a) MMR

Step	Pick	Tokens	New	RVC
0	GB-79894 (M, 60+)	89	V2	0.333
1–11	11 picks	873	—	0.333

(b) LMVU

Step	Pick	Tokens	New	RVC
0	FR-25038 (M, 30-44)	94	V3	0.333
1	MK-27688 (F, 60+)	86	V1	0.667
2	PT-77928 (M, 60+)	71	V2	1.000
3–11	9 picks	731	—	1.000

of 7 viewpoints, while M1's choice (respondent 0517) realizes 4 against an oracle best of 9. At step 0 the selection state is empty, so the state-dependent features carry no signal and M1's ranking error is a pure scoring error. Such states are tallied in Appendix K; they are most frequent on ArgKP, where they are inconsequential, and rarest relative to opportunity on WVS.

M Feature List

The 18 runtime-visible, label-blind features of M1 fall into four groups: candidate descriptors (4), evidence–state geometry (5), atomic–state geometry (5), and shared state descriptors (4). Table 8 gives the full definitions. All geometry is computed from incremental sufficient statistics maintained over the selection—running similarity extrema, the selected-embedding sum, and per-atom maxima—so no pairwise pool-similarity matrix is materialized; an empty selection yields similarity 0 and novelty 1 by convention. M0 uses three candidate descriptors: dense relevance, query–candidate cosine, and candidate token length. Figure 7 profiles the fitted coefficients of the deployed ESS10 models (local training pools, three training seeds), and two readings matter. First, the largest-magnitude weights sit on the evidence–state geometry group, while the static candidate descriptors carry small weights—consistent with the M0 ablation. Second, the four shared state descriptors are constant across candidates within a state: they cancel in the within-state pairwise differences of §4.2 and cannot change the argmax of §4.3, and their fitted coefficients are exactly zero in every fitted model of this version. They are recorded for completeness; the budget binds through the feasibility filter of §4.3, not through the learned score.

N Per-Benchmark Structure

Full selection-pool rows per query (min/mean/max): WVS $506/507.3/508$, Perspectrum $4/8.9/28$, ArgKP $47/75.7/91$, ESS10 $34/88.3/119$. MMR token utilization at budget 1024 is 0.85–0.97 per benchmark (Table 9)—budgets are genuinely binding. Positive-marginal-action frequency under M1: 0.321 (WVS), 0.520 (Perspectrum), 0.110 (ArgKP), 0.404 (ESS10).

O Allocation Dynamics and Headroom

Figure 8 shows mean cumulative RVC against tokens used at budget 1024 on the WVS protected fold and the ESS10 held-out set under this version's full-pool protocol: on WVS, MMR covers faster through the first half of the budget, but LMVU overtakes it at roughly 530 tokens and closes the entire gap to the full-pool oracle (1.000 vs. 0.917); on ESS10 the two track each other for the first $\sim$250 tokens, after which LMVU leads throughout (0.889 vs. 0.843 at the budget).

The sweep of Figure 3 shows that the gain is not tied to one operating point: on ESS10 LMVU improves over MMR by $+0.047$ at 1024 and $+0.040$ at 2048; on Perspectrum LMVU leads MMR at all three budgets while M0 remains strongest at 1024; and on ArgKP the only nonzero gap is

Table 8: The 18 runtime-visible, label-blind features $\phi(q, S, e)$ (max/mean/min variants counted separately), grouped as in Figure 7. S denotes the current selection and u the tokens it uses; z_x is the frozen embedding of unit x; cos is cosine over the frozen embeddings; rel is the dense retriever score; $\bar{z}_S = \frac{1}{|S|}\sum_{e' \in S} z_{e'}$ is the unnormalized mean selected vector; $a \in A(e)$ are the benchmark's atomic sub-evidence units of e, $A(S) = \bigcup_{e' \in S} A(e')$ the selected ones, and $m_a = \max_{a' \in A(S)} \cos(a, a')$.

candidate descriptors	
dense relevance	$\mathrm{rel}(q, e)$
query–candidate cosine	$\cos(q, e)$
candidate token length	$c(e)$
candidate atomic count	$\lvert A(e)\rvert$
evidence–state geometry	
sim. to selected evidence	$\max$ / mean / $\min_{e' \in S} \cos(e, e')$
novelty vs. evidence	$1 - \max_{e' \in S} \cos(e, e')$
candidate centroid sim.	$z_e^\top \bar{z}_S$
atomic–state geometry	
sim. to selected atomic units	$\max$ / $\mathrm{mean}_{a \in A(e)}\, m_a$
novelty vs. atomic units	$1 - \max_{a \in A(e)} m_a$
atomic novelty	mean / $\max_{a \in A(e)} \left[1 - m_a\right]$
shared state descriptors	
selected evidence count	$\lvert S\rvert$
tokens used / remaining	$u,\ B - u$
mean selected relevance	$\mathrm{mean}_{e' \in S}\, \mathrm{rel}(q, e')$

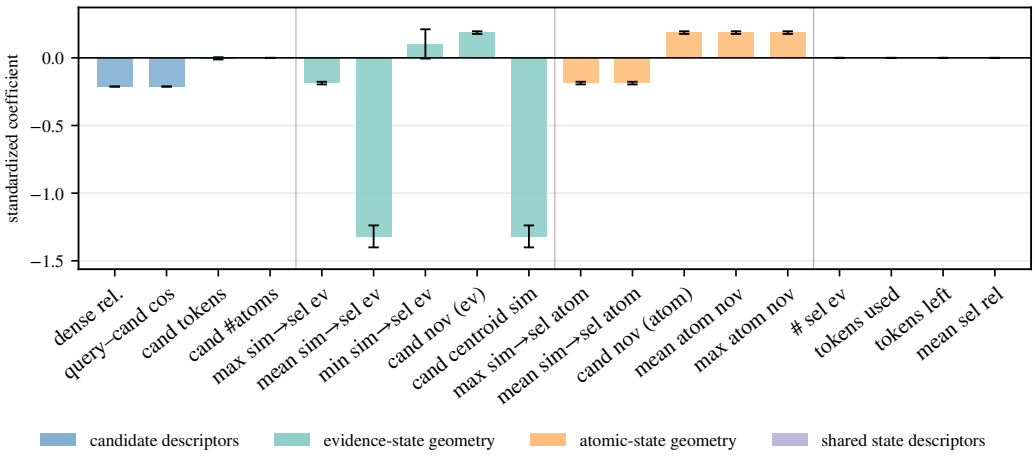


Figure 7: M1 coefficient profile over the 18 runtime-visible features, grouped as in §4; mean ± sd across the three training seeds of the deployed ESS10 model (local training pools; feature labels are abbreviated and presented in the order listed in Appendix M).

+0.0095 at 512. The descriptive correlation (Pearson r=0.574, $n = 4$) links realized gains to available headroom rather than to budget alone.

P Repeated-Support Coverage

This appendix collects the full repeated-support study behind §3, §5, and Table 1(a,b): the source-unit definitions, the audit protocol, the complete budget grid, and the post-hoc objective-alignment ablation that trains on the repeated-support target directly.

Table 9: Structural moderators per benchmark (descriptive; $n = 4$, no causal claim). Headroom $= \mathrm{RVC}_{\mathrm{oracle}} - \mathrm{RVC}_{\mathrm{MMR}}$ at budget 1024 on the full selection pools, and $\mathrm{Headroom}_2$ the same difference under RVC_2; rows/viewpoint measures full-pool redundancy; MMR util. is MMR's mean token utilization at budget 1024. Cell shading scales with value within each column (LMVU − MMR shaded red by magnitude, zero unshaded).

Benchmark	Headroom	$\mathrm{Headroom}_2$	LMVU − MMR	Rows/viewpoint	Viewpoints/query	MMR util.
WVS (protected)	0.083	0.218	+0.083	84.6	6.0	0.931
Perspectrum	0.069	0.074	+0.016	3.0	3.0	0.853
ArgKP	0.000	0.000	0.000	13.8	5.5	0.968
ESS10	0.157	0.206	+0.047	18.4	4.8	0.961

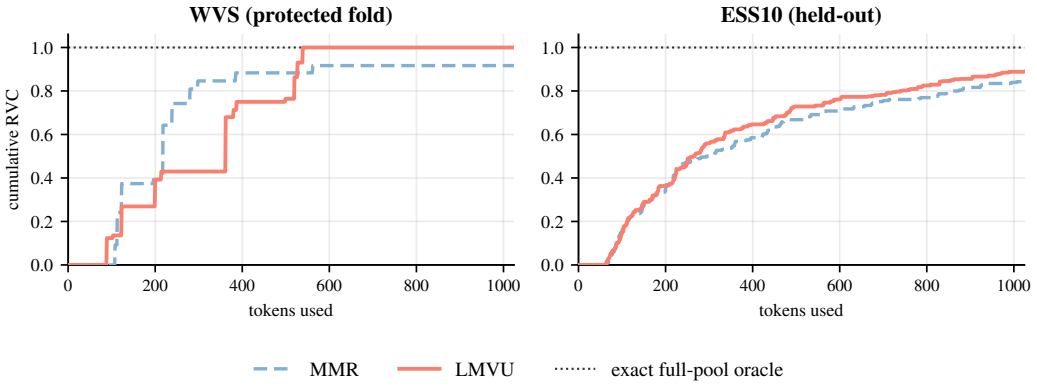


Figure 8: Mean cumulative RVC vs. tokens used at budget 1024 under the full-pool protocol (WVS protected fold; ESS10 held-out): the LMVU local-training arm (three seeds, seed means within query, macro-averaged) against deterministic full-pool MMR; the dotted line represents the exact full-pool oracle, which has RVC equal to 1.0 on each of the two benchmarks.

The RVC_n family. For the evaluation family in §5, RVC credits a viewpoint once, whereas higher thresholds retain credit for additional distinct source units. Writing the source count explicitly,

$$\mathrm{RVC}_n(S) \;=\; \frac{1}{|V_q|}\sum_{v\in V_q} \min\Big\{\frac{n_v(S)}{n},\,1\Big\}, \qquad n_v(S) \;=\; \big|\{\,\mathrm{src}(e) : e \in S,\; v \in V(e)\,\}\big|,$$

where $\mathrm{src}(e)$ maps an passage to its source unit. RVC_n is monotone in S and nonincreasing in n, and n=1 recovers RVC exactly. We report n=1 (breadth) and n=2, the minimal extension beyond singleton coverage: RVC_2 is one precisely when every valid viewpoint is backed by at least two distinct sources. The repeated-support ceiling replaces the oracle objective by $\sum_{v\in V_q}\min\{n_v(S), 2\}$ and is solved exactly by a base-3 (trit-state) dynamic program over per-viewpoint source counts on the full selection pools $E(q)$ (Proposition 3), giving $S_q^{\star(2)}$ and $\mathrm{RVC}_2(S_q^{\star(2)})$, with $\mathrm{headroom}_2 = \mathrm{RVC}_2(S_q^{\star(2)}) - \mathrm{RVC}_2(\mathrm{MMR})$ defined analogously to headroom.

Source units. Table 10 lists what $n_v(S)$ counts distinct instances of on each benchmark: a survey respondent on WVS and ESS10, one claim-scoped document on Perspectrum, and the individual argument itself on ArgKP. Parsing is total—every item in the full pools maps to exactly one unit (0 parse failures, verified per query–budget cell in the oracle audit)—and within every query distinct pool items carry distinct units, the injectivity hypothesis of Proposition 3. Two cautions apply. First, repeated support is not independent corroboration: Perspectrum's documents trace to only three source websites, so two documents supporting one viewpoint may share an origin; we report the document-level unit as primary and make no site-independence claim. Second, RVC_2 is bounded by RVC trajectory-wise (a second unit implies a first) and held on every scored trajectory.

Table 10: Source units for RVC_2 per benchmark: what $n_v(S)$ counts distinct instances of. Pool items and distinct units are totals over the full selection pools; within each query the item-to-unit mapping is injective. Shading scales with value within column.

Benchmark	Source unit	Pool items	Distinct units
WVS (protected)	survey respondent	3044	1522
Perspectrum	claim-scoped document	859	859
ArgKP	individual argument	454	454
ESS10	survey respondent	3003	2290

Audit protocol of the full-pool extension. All trajectories come from two gated sources (Appendix A): the delivered LF matrix (integrity-verified, bit-reproduced by parameter replay) and the full-pool baseline rerun (gated on per-query reproduction of the delivered MMR and Dense rows at tolerance 10^{-9}). Every run carries both RVC and RVC_2 under the source-unit definitions above. The exact repeated-support oracle was computed once per query–budget cell on the full pools and verified by an in-process identical second run; it dominates every method's RVC_2 on every query (0 violations) and never exceeds the breadth oracle. As a cross-check, the same construction restricted to the original local pools reproduces the locked local-pool repeated-support audit exactly on all twelve dataset–budget cells. For the protected WVS fold this audit reused the same gold annotations read-only; no method, tuning, or training state changed.

Full budget grid. Table 11 reports RVC_2 for all methods and the exact repeated-support oracle under the full-pool protocol. WVS is tabulated at the primary budget only, matching the main-text reporting scope; the other benchmarks span all three budgets.

Table 11: Repeated-support coverage RVC_2 across budgets (macro mean per query, same protocol as Table 1); WVS at the primary budget only. Best non-oracle per column in bold; shading scales with RVC_2 within column (oracle row excluded). Yellow rows are the training-heavy neural baselines (JPR, RichRAG); they are excluded from the bold comparison and from the per-column shading scales.

	WVS	Perspectrum			ArgKP			ESS10		
	1024	512	1024	2048	512	1024	2048	512	1024	2048
Random	0.8757	0.5296	0.7932	0.9491	0.8651	0.9720	1.0000	0.4935	0.7223	0.8975
BM25	0.6583	0.5083	0.7720	0.9276	0.7952	0.9881	1.0000	0.4827	0.6769	0.8657
FacLoc	**0.9167**	0.5370	0.8045	0.9419	0.7863	0.9881	1.0000	0.5036	0.6781	0.8583
xQuAD	0.7114	0.5477	0.8083	0.9426	0.8450	0.9733	1.0000	0.4474	0.6422	0.8348
kDPP	0.9042	0.5518	0.7995	0.9503	0.9143	1.0000	1.0000	0.5043	0.7047	0.9032
SubMix	0.6898	0.5404	0.8002	0.9397	0.8310	0.9714	1.0000	0.4515	0.6173	0.8283
LexRank	0.7315	0.5411	0.7986	0.9389	0.8310	0.9881	1.0000	0.4566	0.6220	0.8344
AdaGReS	0.6148	0.2904	0.5018	0.7692	0.7548	0.9143	1.0000	0.3058	0.5138	0.7026
MMKP	0.3120	0.6078	0.8329	0.9465	0.8649	0.9173	0.9173	0.1251	0.1251	0.1251
Dense	0.6898	0.5432	0.8014	0.9397	0.8310	0.9714	1.0000	0.4515	0.6154	0.8283
CeRerank	0.6704	0.5230	0.7923	0.9353	0.7774	0.9458	1.0000	0.3933	0.5285	0.7149
Cluster	0.7753	0.5488	0.7999	0.9508	0.8512	0.9631	1.0000	0.4710	0.6960	0.8781
MMR	0.7824	0.5471	0.8160	0.9500	0.8929	1.0000	1.0000	0.5312	0.7273	0.8778
M0	0.7444	**0.6080**	**0.8480**	**0.9668**	0.7988	0.9131	1.0000	0.4980	0.7206	0.8559
LMVU	0.9015	0.5753	0.8373	0.9640	**0.9310**	**1.0000**	**1.0000**	**0.5532**	**0.7790**	**0.9343**
JPR	0.8097	0.5402	0.7996	0.9548	0.9921	1.0000	1.0000	0.6075	0.6839	0.7511
RichRAG	0.7994	0.5151	0.7906	0.9302	0.7972	0.9234	1.0000	0.5442	0.8315	0.9674
Oracle_2	1.0000	0.6320	0.8898	0.9907	1.0000	1.0000	1.0000	0.6612	0.9334	1.0000

The strongest depth allocator depends on the benchmark: M0 leads Perspectrum at all three budgets, FacLoc edges LMVU on WVS at 1024 (0.9167 vs. 0.9015), and on ESS10 LMVU leads the non-neural field throughout the grid, while the training-heavy RICHRAG (yellow row, outside the bold comparison) surpasses it at 1024 and 2048 (0.8315 vs. 0.7790; 0.9674 vs. 0.9343) but not at 512 (0.5442 vs. 0.5532). Budget changes also have different effects. LMVU's RVC_2 advantage over MMR narrows from 0.0282 to 0.0140 on Perspectrum but grows from 0.0220 to 0.0565 on ESS10. Higher absolute coverage need not mean a smaller gap to the achievable ceiling: on ESS10, the remaining Oracle_2–LMVU gap is 0.108, 0.154, and 0.066 at 512, 1024, and 2048 to-

kens. Unlike the MMR-based $\mathrm{headroom}_2$, this gap measures LMVU's remaining shortfall against a budget-dependent ceiling.

Breadth saturation can conceal incomplete repeated support. On WVS at 1024, LMVU reaches RVC 1.000 but RVC_2 only 0.9015, below Oracle_2 1.000. ArgKP is not fully breadth-saturated at 512: LMVU and MMR attain RVC 0.9762 and 0.9667, respectively. Its depth gap is larger (Oracle_2 1.000 vs. MMR 0.893), and LMVU captures part (0.931, $+0.038$ over MMR). Both LMVU and MMR reach the breadth and depth ceilings at 1024 and 2048, explaining their zero differences at those budgets rather than establishing equivalence away from saturation.

The mean depth deficits at 512 tokens on WVS and ESS10 under the original local-selection protocol are absent from these full-pool comparisons. LMVU improves both metrics over MMR on ESS10 (0.553 vs. 0.531 for RVC_2; Table 11). The same depth ordering holds on WVS (0.735 vs. 0.698), an existing 512-token result not tabulated here; its breadth gain is also positive. These observations describe the full-pool runs, not a new protected-fold evaluation or a general guarantee that breadth and depth objectives agree across datasets and budgets.

Paired uncertainty. Table 12 reports paired LMVU–MMR differences for the same tabulated budgets. These exploratory statistics distinguish positive macro means from consistent query-level gains, retaining ties and losses rather than treating every query as improved.

Table 12: LMVU − MMR paired statistics: $\Delta\mathrm{RVC}$ at the primary budget 1024 (top block) and $\Delta\mathrm{RVC}_2$ across budgets (bottom block); WVS at the primary budget only. Intervals are bootstrap 95% CIs, p the two-sided sign test; WVS/ArgKP statistics are descriptive (three groups). Positive Δ cells shaded by magnitude.

Metric	Benchmark	Budget	Δ	95% CI	w/t/l	p
RVC	WVS	1024	+0.0833	—	2/4/0	1.00
	Perspectrum	1024	+0.0161	[−0.001, +0.035]	11/82/4	0.12
	ArgKP	1024	0.0000	—	0/6/0	1.00
	ESS10	1024	+0.0466	[−0.022, +0.118]	10/18/6	0.45
RVC_2	WVS	1024	+0.1191	—	4/1/1	1.00
	Perspectrum	512	+0.0282	[+0.014, +0.044]	19/76/2	0.0002
		1024	+0.0213	[+0.007, +0.037]	16/75/6	0.052
		2048	+0.0140	[+0.004, +0.025]	11/83/3	0.057
	ArgKP	512	+0.0381	—	3/2/1	1.00
		1024	0.0000	—	0/6/0	1.00
		2048	0.0000	—	0/6/0	1.00
	ESS10	512	+0.0220	[−0.016, +0.065]	7/22/5	0.77
		1024	+0.0518	[−0.000, +0.109]	15/11/8	0.21
		2048	+0.0565	[+0.009, +0.107]	12/17/5	0.14

The breadth block at the primary budget is nonnegative everywhere (ArgKP ties at the ceiling) and no interval excludes zero. On Perspectrum, all three bootstrap intervals exclude zero, but only the 512-token sign test has $p < 0.05$; at 2048, 83 of 97 queries tie. On ESS10, all mean differences are positive, yet only the 2048-token bootstrap interval excludes zero and none of the sign tests has $p < 0.05$. The 1024-token lower bound is slightly negative, displayed as -0.000 after rounding. These summaries answer different questions: the bootstrap interval retains the magnitudes of query-level differences, whereas the sign test uses their directions after excluding ties. Thus an interval above zero need not imply that wins substantially outnumber losses among the queries that change.

For WVS and ArgKP, the reported win/tie/loss counts describe six queries, but the descriptive sign tests operate on three issue or topic groups. The group count is too small for the CI reporting used on the other benchmarks. In particular, WVS's positive mean gain is not evidence that every protected query improves on depth. Across benchmarks, positive means therefore coexist with unchanged queries and occasional regressions under the same budget.

Objective-alignment ablation (post hoc). We next isolate the training target under the original local-selection protocol (Appendix A), separately from the full-pool results above. LMVU's training

target is the breadth gain ΔV; a natural question is whether pointing the same construction at depth helps RVC_2 directly. The repeated-support realized gain of a candidate in state (q, S) is

$$\Delta_2(e \mid q, S) \;=\; \frac{1}{2\,|V_q|}\;\#\big\{\, v \in V(e) \;:\; n_v(S) < 2 \text{ and } \mathrm{src}(e) \notin \mathrm{sources}_v(S) \,\big\},$$

exactly the realized increment of RVC_2 (a viewpoint gains $1/(2|V_q|)$ for its first and again for its second distinct supporting unit). We retrained the unchanged ranker class (linear, the same 18 features, C=1.0, 20-pair cap, seed 41) on the canonical development states with Δ_2 targets on ESS10 and Perspectrum, and evaluated the frozen model on the same held-out folds. The protected WVS fold was untouched; ArgKP was excluded as saturated at the primary budget. This post-hoc arm is an objective-alignment probe, not a candidate method; no claim rests on it.

Table 13: Objective-alignment ablation: the canonical M1 ranker vs. the same construction trained on Δ_2 ($\mathrm{M1}_{\Delta_2}$), with paired $\Delta\mathrm{RVC}_2$ statistics (bootstrap 95% CI; sign-test p). Post-hoc arm on held-out folds under the original protocol (Appendix A); positive Δ cells shaded by magnitude.

		RVC			RVC_2			
Benchmark	Budget	M1	$\mathrm{M1}_{\Delta_2}$	Δ [95% CI]	M1	$\mathrm{M1}_{\Delta_2}$	Δ [95% CI]	p
Perspectrum	512	0.8134	0.8292	+0.0157 [−0.007, +0.041]	0.5751	0.6043	+0.0292 [+0.013, +0.047]	0.001
	1024	0.9392	0.9563	+0.0172 [+0.002, +0.034]	0.8286	0.8509	+0.0224 [+0.004, +0.041]	0.004
	2048	0.9901	0.9933	+0.0032 [−0.006, +0.013]	0.9628	0.9678	+0.0051 [−0.006, +0.017]	0.774
ESS10	512	0.6707	0.6962	+0.0255 [−0.029, +0.083]	0.5170	0.5272	+0.0102 [−0.019, +0.039]	0.388
	1024	0.8866	0.8678	−0.0187 [−0.051, +0.012]	0.7742	0.7610	−0.0133 [−0.038, +0.009]	0.549
	2048	0.9739	0.9815	+0.0076 [−0.010, +0.027]	0.9284	0.9348	+0.0064 [−0.014, +0.026]	0.549

Table 13 shows that target alignment helps under some, but not all, conditions. On Perspectrum, Δ_2 training improves RVC_2 by $+0.029$ at 512 and $+0.022$ at 1024, with CIs excluding zero and sign-test $p = 0.001, 0.004$. At 1024, RVC also increases by $+0.017$, with a positive bootstrap interval: improved depth need not come at the expense of breadth in this setting. At 2048, the smaller depth gain of $+0.005$ has an interval crossing zero, so the lower-budget evidence does not establish a reliable benefit throughout the budget range.

On ESS10, all three depth intervals cross zero. At 1024, both RVC and RVC_2 decline in mean (-0.019 and -0.013), with neither interval excluding zero. This is not an observed tradeoff in which depth improves while breadth deteriorates; nor does it identify redundancy as a causal explanation. The experiment changes the target, but does not guarantee that the same ranker will learn a better selection policy for that target. These comparisons concern canonical M1 and $\mathrm{M1}_{\Delta_2}$ within the original protocol, not the full-pool LMVU scores in Table 11.

Together, the full-pool evaluation and the separate target probe show why breadth and repeated support should be reported together. A breadth ceiling can leave depth unresolved, while directly training for depth does not yield a uniform improvement across datasets and budgets. These are selection-level findings: distinct source units need not provide independent corroboration, and neither study establishes an improvement in end-to-end generation quality.