WARP: Wasserstein-Aligned RAG for Population Opinions

Aman Singh Thakur* **Aditya Agrawal** **Alwarappan Nakkiran** **Alex Karlsson**

Amazon.com

Abstract

RAG systems are increasingly used to summarize what large collections of documents say. A user asks "What do people think about X?" and receives an answer that reads as consensus. But standard top-k retrieval ranks documents by query similarity, not by how faithfully they represent the population, so minority views quietly disappear. Existing fixes fall short. Diversity re-rankers like MMR and DPP spread retrieved documents apart, but with no target distribution to aim for. Calibration methods based on KL or JS divergence do target one, yet treat opinion bins as unordered: confusing strong positive with strong negative costs no more than an adjacent-bin miss.

We introduce WARP, a family of post-retrieval algorithms that calibrate retrieved evidence to the population's opinion distribution. WARP first recovers underrepresented opinions that cosine ranking may bury, then uses Wasserstein-1 distance to select documents whose sentiment-intensity distribution matches the population target, capturing the ordinal structure ignored by KL and JS divergence. We develop three variants for dense, sparse, and variable candidate pools, trading off calibration quality and speed. Across three review domains spanning 35K documents, 156 queries, and 26 entities, WARP's domain-matched variants reduce distributional error by at least 43% with sub-second latency. These gains carry through to generation: a five-judge LLM panel prefers WARP-generated answers in up to 86% of decided comparisons at k=5.

1 Introduction

Users used to browse reviews themselves - clicking through results, weighing contradictions, building a mental picture. Noisy, but transparent: you could see disagreement. RAG systems (Lewis et al., 2020) replace this with a single synthesized answer, but a new failure mode emerges. Standard top-k retrieval selects evidence by query similarity, without factoring spread of opinions in the corpus. Take a hotel where 60% of reviewers are satisfied and 40% complain about noise. Retrieval over-represents the majority because similarity scoring favors the dominant cluster. Every claim in the resulting summary traces to a real review, yet the user never learns that two in five guests had a bad experience. This leads to distributional distortion. A synthesized response from a skewed sample would read factual but would not present the whole picture.

For opinion queries, uncertainty is aleatoric: it reflects genuine disagreement. A faithful system must preserve the distribution without collapsing it. Agrawal et al. (2026) formalize this distinction; Nayeem and Rafiei (2025) build a scalable opinion summarization pipeline but do not optimize the retriever for distributional fidelity. **What's missing is the engineering: in a production stack with sub-second latency budgets, how do you select a small evidence set whose opinion distribution actually matches the population?**

A standard cosine retriever maximizes query similarity, producing relevant but homogeneous evidence - four glowing reviews when two would suffice. Diversity re-rankers (MMR (Carbonell and Goldstein, 1998), DPP (Kulesza and Taskar, 2012)) push back by maximizing pairwise spread, which helps until each sentiment bin has one representative; past that point they quietly revert to relevance ordering. KL and JS calibration (Steck, 2018; Dang and Croft, 2012) come closer by explicitly optimizing against a target distribution. But they remain categorical. Confusing "positive" with "negative" costs the same as confusing "positive" and "neutral". Opinions are ordinal and we need a metric which supports this.

Wasserstein-1 respects this ordering. Its closed-form CDF computation ($O(m)$ for m sentiment bins) makes per-candidate evaluation cheap enough

* Correspondence: amanzing@amazon.com

EMNLP 2026 Industry Track, pages 2684–2704
October 24-29, 2026 ©2026 Association for Computational Linguistics

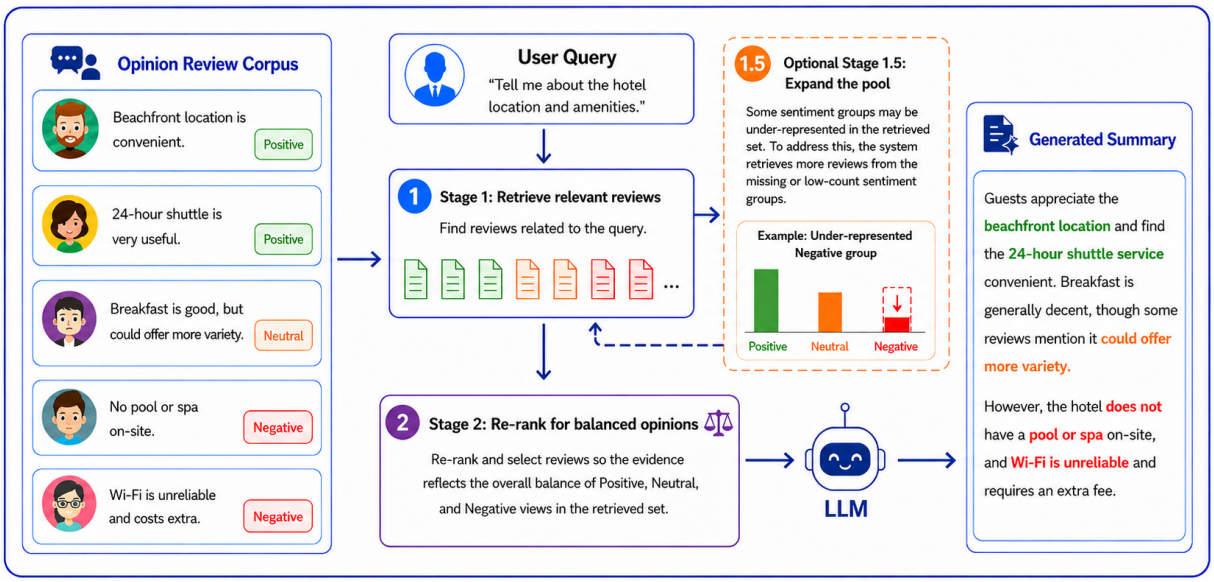

Figure 1: WARP pipeline. **Stage 1**: semantic retrieval returns a Top-N candidate pool. **Stage 1.5**: deficit-aware pool expansion recovers under-represented sentiment poles via re-retrieval (*Entity-Gated* or *Adaptive Expansion*; §3.3; self-bypasses when the pool is already balanced). **Stage 2**: W_1 re-ranking selects the final k documents whose empirical opinion distribution matches the population target P_{pop} (W_1 *Minimizer* for dense pools, W_1*-MMR* for sparse pools, or *WassRank OT* as a tuning-free fallback; §3.4). Pool density drives the variant choice (decision matrix in Table 3).

for runtime re-ranking without offline precomputation. We built WARP (Figure 1) around this in two stages. First, a deficit-aware pool expansion pass compares the retrieved distribution against the population target, identifies under-represented poles, and issues targeted re-retrievals only where gaps exist. Then a greedy re-ranker selects the final evidence set by minimizing W_1 distance to P_{pop}. The pipeline instantiates in three variants (Section 3.4): the Minimizer greedily selects from entity-matched candidates - effective for dense pools; for sparser conditions, W_1-MMR blends relevance with calibration, and WassRank OT solves the full slot-assignment as rectangular optimal transport requiring no per-domain tuning.

Our contributions:

1. **Pool expansion that recovers what cosine retrieval buries.** Entity-gated and adaptive re-retrieval recover under-represented opinions with near-complete entity coverage. Self-bypasses at zero cost when already balanced.

2. **W_1 re-rankers substantially reduce distributional error.** A greedy minimizer that matches opinion distribution for entity dense domains and the hybrid variants trade opinion diversity for relevance on sparse domains.

3. **Deployment characterization across 3 domains.** Pool density determines algorithm choice. All variants add under 310 ms re-ranking latency (p99) and generalize across labeling methods and index types.

2 Related Work

Retrieval diversity. MMR (Carbonell and Goldstein, 1998) penalizes redundancy via pairwise dissimilarity; DPP (Kulesza and Taskar, 2012) maximizes determinantal spread; xQuAD (Santos et al., 2010) diversifies over query intents. More recent variants-BQP (Lu and Sidiropoulos, 2026), Ada-GReS (Peng et al., 2025), MUSS (Nguyen and Kan, 2026)-refine these ideas in various ways (see Wu et al. (2024) for a survey). However, diversity is the wrong objective here as the signal runs out, once every sentiment bin has one representative (5–7 selections).

Optimal transport (OT) in IR. Wasserstein distances (Peyré and Cuturi, 2019) have appeared in IR before-Word Mover's Distance (Kusner et al., 2015) for document similarity, WassRank (Yu et al., 2019) as a listwise training loss, OTExtSum (Tang et al., 2022) for extractive summarization, Wasserstein coresets (Claici et al., 2018) for data summarization. However, all these function at training time, over generic content. To our knowledge, none

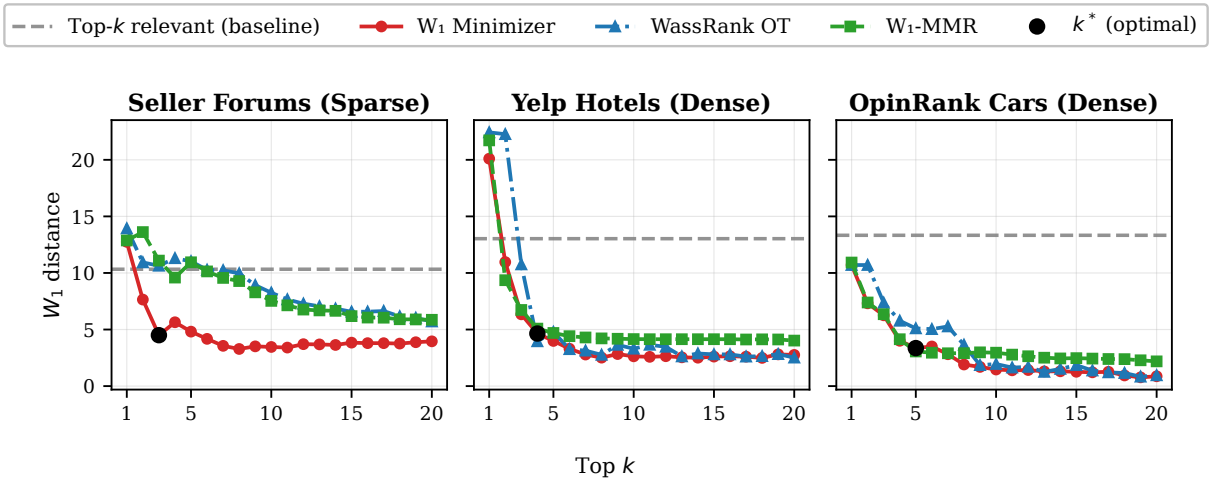

Figure 2: W_1 re-rankers (no re-retrieval): domain-matched variants reduce distributional error at least 43% vs. Top-k (gray dashed) — 43% Seller Forums (W_1-MMR), 80% Yelp (W_1 Minimizer), 69% OpinRank (W_1 Minimizer); see Table 1. Curves converge at k^*=3–5 documents (black dot; Appendix C.3) as most entities' opinion mass concentrates in 2–3 of 7 bins. N=200.

use Wasserstein as a *runtime* selection objective conditioned on an opinion distribution.

Opinion-aware retrieval. Agrawal et al. (2026) audit 30+ RAG benchmarks and find a striking gap: only one addresses opinion synthesis. They formalize the epistemic/aleatoric distinction - the posterior converges to the population distribution $P_{\text{pop}}(\theta)$ for opinion queries - and propose a three-term objective combining coverage (W_2), fidelity, and demographic fairness. However, their work is limited to 2 domains without practical retrieval strategies. Nayeem and Rafiei (2025) solve a different problem entirely: generating opinion highlights from thousands of reviews via retrieve-then-synthesize with AOS-triplet verification. Their retriever is standard semantic search without targeting distributional fidelity. We instantiate the Agrawal et al. (2026) coverage objective using W_1 instead of W_2 - its closed-form CDF difference enables $O(m)$ per-candidate evaluation at runtime - and test across three domains.

Calibrated selection and diversity methods. Calibrated recommendation via KL minimization (Steck, 2018) and proportional slot allocation (PM-2; Dang and Croft 2012) do address distributional fidelity across facets - but assume categorical labels. Fairness-aware ranking (Singh and Joachims, 2018; Zehlike et al., 2017; Oesterling et al., 2024) enforces demographic constraints without modeling ordinal opinion at all. On the sentiment side, Aktolga and Allan (2013) applies bias modes to retrieval, aspect-level summarization (Angelidis and Lapata, 2018; Jiang et al., 2023) diversifies across topical facets. The methods we surveyed all use categorical penalty structures that do not exploit ordinal distance between opinion bins. WARP addresses this gap: an ordinal W_1 ground cost applied at query time to opinion-distribution matching in RAG, drawing on the calibrated-selection tradition (Steck, 2018) and OT-in-ranking (Yu et al., 2019) while adapting both to a training-free runtime re-ranker across three review domains.

3 Methodology

Below we describe the experimental setup (Sections 3.1–3.2), pool expansion (Section 3.3), and W_1 re-ranking (Section 3.4) for stress-testing WARP.

3.1 Datasets

3 review corpora give us the spread we need across different domains:

- **Amazon Seller Forums**[1] (∼8K documents). Retrieved via Amazon Bedrock Knowledge Bases (Amazon Web Services, 2024b) with hybrid search.
- **Yelp Open Dataset** (Yelp, 2024) (∼14K hotel reviews). Retrieved via local FAISS (Johnson et al., 2019) + MiniLM-L6 (Wang et al., 2020; Reimers and Gurevych, 2019).

[1] https://sellercentral.amazon.com/seller-forums

- **OpinRank** (Ganesan and Zhai, 2012) ($\sim$13K automotive reviews). Retrieved via local FAISS + MiniLM-L6.

Entity selection follows Agrawal et al. (2026): LLM-based entity extraction, diversity filtering (Shannon entropy $H \geq 0.6$, minority sentiment $\geq$10%), then ranking by entropy.

We use six questions per entity (2 breadth, 2 polar, 2 segment; templates in Appendix A.2) which gives us 156 total queries, across all domains. Entity and sentiment-intensity labels come from a single LLM pass per document (prompt in Appendix A.3.1), mapped to a 7-bin ordinal scale $\mathcal{S} = \{-30, -20, -10, 0, +10, +20, +30\}$ (Table 4, Appendix A.1).

Full dataset statistics appear in Table 5 (Appendix A.1).

Defaults: N=200 candidate pool, k=20 output.

3.2 Baselines

Five baselines. Top-k is plain cosine-similarity retrieval. MMR (Carbonell and Goldstein, 1998) penalizes semantic redundancy (λ=0.5). DPP (Kulesza and Taskar, 2012) maximizes determinantal spread over 1D sentiment-intensity features. OpinionMMR penalizes opinion-bin distance rather than embedding distance. KL (JS) Minimizer implements Steck (2018)'s calibrated-recommendation objective with Jensen–Shannon divergence - identical to our W_1 Minimizer but with an order-agnostic metric (Appendix C.1).

Metrics. Distributional fidelity: Wasserstein-1 distance W_1 (lower is better). Entity relevance: Entity Match rate (EM%, hereafter EM; fraction of selected documents matching the queried entity, higher is better). We also report re-ranking latency in milliseconds. For generation evaluation, Cohen's κ (Cohen, 1960) measures inter-judge alignment.

3.3 Pool Expansion via Re-Retrieval

For entity sparse domains, the relevance-based retrieval step typically returns a substantial fraction of documents that do not discuss the queried entity - due to near-misses leaked in from other entities sharing the same index. We tackle this with two expansion strategies (full algorithms in Appendix B):

Entity-Gated Re-retrieval. Pass 1 pulls the standard top-N by relevance. Pass 2 then digs into the tail of the ranked list, fishing out documents that match the queried entity but fell below the cutoff, ranking them by opinion extremity (absolute sentiment-intensity score - strongly negative reviews surface before mild ones). The passes are merged and deduplicated.

Algorithm 1 W_1 Minimizer

Require: Pool C, entity e, target P_{pop}, output size k

$C_e \leftarrow \{d \in C : \text{entity}(d) = e\}$
$S \leftarrow \emptyset$
while $|S| < k$ **do**
  $d^* \leftarrow \arg\min_{d \in C_e \setminus S} W_1(\text{EmpDist}(S \cup \{d\}), P_{\text{pop}})$
    ties broken by relevance score $\downarrow$
  $S \leftarrow S \cup \{d^*\}$
end while
return S

Adaptive Expansion. We compare the pool's opinion breakdown against P_{pop}. If any bin is under-represented (pool fraction below threshold of the population fraction), a targeted retrieval fires for entity-matched documents in the deficit bin using pole-biased queries (Prompt A.3.4.1). If the pool already mirrors the target, nothing happens - zero overhead (Algorithm 3).

3.4 Wasserstein Re-Ranking Algorithms

Problem Formulation. For opinion queries, a faithful answer must reflect how viewpoints distribute along an ordinal axis - one where positions have a natural ordering and distances between them reflect severity of disagreement. The re-ranking objective follows directly: select k documents whose empirical distribution along this axis minimizes transport distance to the target distribution P_{pop}. We instantiate the axis as sentiment-intensity (SI), a 7-bin ordinal scale $\mathcal{S} = \{-30, -20, \ldots, +30\}$. At indexing time each document d receives an SI label; per entity e we derive $P_{\text{pop}}(s)$ as the corpus-observed fraction of documents for e in each bin, or as an externally-specified target (star ratings, survey-calibrated priors) when one is available — P_{pop} is an input to the re-ranker, not a claim about the true underlying population (§Limitations; tolerance to partial estimates in Section 4.4). Given a candidate pool C of N documents, pick $k \ll N$ into result set S minimizing:

$$W_1(P, Q) = \sum_{i=1}^{m-1} |\text{CDF}_P(s_i) - \text{CDF}_Q(s_i)| \cdot \Delta s \quad (1)$$

Table 1: Main results: Diversity baselines and calibrated methods vs WARP re-rankers. $W_1\downarrow$: distributional distance to the population opinion target. EM%↑: fraction of selected documents matching the queried entity. EG/AE = pool expansion strategies (Entity-Gated / Adaptive Expansion). N=200 candidates, k=20 output. Paired Wilcoxon vs. Top-k: $^{***}p<0.001$, $^{**}p<0.01$, $^{*}p<0.05$. Bold: best per column. †Shared FAISS index across all OpinRank entities (EM=82.9%); entity-specific indices give W_1=0.95 with 100% EM (Appendix F.3).

	Seller Forums		**Yelp Hotels**		**OpinRank Cars**		**Latency**
Method	$W_1\downarrow$	EM%	$W_1\downarrow$	EM%	$W_1\downarrow$	EM%	**(ms)**
Top-k (baseline)	10.33	42.9	13.60	40.1	13.33	27.0	**0**
MMR*	10.31	43.5	8.67	14.2	9.64	11.3	12
DPP***	11.69	41.2	5.63	78.8	7.38	69.2	45
OpinionMMR**	10.44	43.5	8.59	12.4	10.02	10.6	12
KL (JS) Minimizer*	8.87	51.4	3.00	93.8	4.54	85.8	**9**
W_1-MMR***	5.86	44.6	4.02	19.4	**3.41**	21.8	154
W_1 Minimizer***	8.84	51.4	2.76	93.8	4.14	**85.8**	**9**
WassRank OT***	**5.71**	44.4	2.52	92.2	3.87†	82.9	89
EG + W_1 Min***	8.59	**63.5**	**1.52**	**99.2**	4.14	85.8	13
AE + W_1 Min***	8.59	**63.5**	1.66	95.9	4.14	85.8	**8**

where CDF is the cumulative distribution function, m=7 bins, and Δs=10 is the bin spacing. The ordinal structure matters: confusing $+30$ with -30 costs $W_1^{\max}$=60, while adjacent-bin errors cost only 10. Evaluation runs in $O(m)$ per candidate; we observe empirically that rank ordering is preserved under W_2 (Appendix C.4).

We implemented three variants that minimize this objective. They differ in how they cope with entity pool density and the relevance-calibration tradeoff:

W_1 Minimizer. Filter the pool to entity-matched candidates, then greedily add whichever document pulls the empirical distribution closest to P_{pop}, breaking ties by relevance score (Algorithm 1). Needs a dense pool where most candidates match the queried entity.

W_1-MMR. Built for sparse pools where entity-matched candidates alone cannot fill k slots. This variant works on the full candidate set - no entity filtering - and scores each candidate by blending retrieval relevance with the calibration gain it would contribute:

$$\text{score}(d) = \lambda \cdot \text{rel}(d) + (1-\lambda) \cdot \frac{\Delta W_1}{W_1^{\text{cur}}} \qquad (2)$$

$\text{rel}(d)$ is the original retrieval score, λ controls the relevance-calibration blend, and dividing by W_1^{cur} normalizes to percentage improvement so the calibration term does not vanish as the distribution tightens. Full algorithm in Appendix B.3.

WassRank OT. Instead of greedy selection, WassRank (Yu et al., 2019) solves the assignment globally: allocate k slots proportionally to P_{pop}, then find the minimum-cost candidate-to-slot bijection where cost blends normalized ordinal distance with a relevance penalty. This minimizes blended transport cost to a proportional discretization of P_{pop}, which empirically tracks W_1 (Appendix F.3). We solve the rectangular assignment (Crouse, 2016) in $O(k^2 \cdot N)$. Originally a listwise training loss; we repurpose it here as an inference-time re-ranker with no learned parameters and call it WassRank OT.

4 Results

Four questions drove our experiments: does it work, does the metric choice matter, do retrieval gains actually reach the user, and which variant belongs where? Sections 4.1–4.4 take them in order.

4.1 Main Results

In Table 1, the gains are large and consistent, concentrating on entities with the most skewed baseline distributions (per-entity breakdowns in Appendix C.5). Our domain-matched W_1 re-ranking algorithms (Section 3.4) cut distributional error (Equation 1) by at least 43% relative to Top-k — 43% on sparse Seller Forums (W_1-MMR), 80% on dense Yelp (W_1 Minimizer, rising to 89% with pool expansion), and 69% on OpinRank (W_1 Minimizer). Entity relevance tracks along: pairing Entity-Gated re-retrieval (Section 3.3) with W_1 Minimizer pushes Yelp EM% from 40.1% (Top-k baseline) to 99.2% (post-reranking). As a sanity check, 5% Gaussian perturbation of retrieval scores yields only 2–5% W_1 improvement, non-

significant. This confirms that the score structure drives these results and not the arbitrary reshuffling. (Appendix F.2)

4.2 The Ordinal Metric Makes a Measurable Difference

The metric matters. Swapping JS divergence for W_1 while keeping the same greedy loop and entity filtering cuts error by 8.7% on Yelp and 9.7% on OpinRank (Appendix C.1). JS penalizes a $+30 \leftrightarrow -30$ confusion no more than an adjacent-bin slip; W_1 charges six times more. Entity filtering alone accounts for 45.7% W_1 reduction on Yelp; W_1 Minimizer adds 62.6% beyond that, and on entity sparse domains, like Seller Forums, W_1-MMR adds 37.7% beyond the entity-filter baseline (Appendix C.2). We also stress-tested under noise: corrupting both SI labels and P_{pop} simultaneously, W_1 Minimizer never falls below Top-k even at 50% label error on dense pools, and still outperforms Top-k by 72% at 20% Dirichlet perturbation (Appendix F.1).

4.3 Retrieval Gains Propagate to Generation

Do retrieval gains actually reach users? We generated answers from each method's retrieved evidence (prompt in Appendix A.3.2) and ran blind pairwise comparisons against Top-k using a 5-judge LLM panel (Claude Sonnet 4 (Anthropic, 2025), Llama 3.3 70B (Meta, 2024), Mistral Large 3 (Mistral AI, 2025), Amazon Nova Pro (Amazon AGI, 2025), DeepSeek v3.2 (DeepSeek-AI, 2025)) with position control (Zheng et al., 2023) and 3/5 majority vote (Appendix G.1). All five judges from independent model families agree directionally at κ_d=0.61–1.00; the random baseline correctly shows no significance.

At k=5, calibration methods win 70–89% of decided comparisons (Table 2). At k=10 this drops to 73–88% (Table 20) as larger context lets the LLM average out distributional imbalance (Appendix G.1.3). A prompt ablation (Appendix G.1.4) disentangles two evaluation criteria: when judges assess *proportional accuracy* (does the answer reflect how common each view is?), W_1 Minimizer leads DPP by 8 pp and matches an oracle stratified-sampler within 1 pp (Table 21); under pure *viewpoint coverage* (does it mention all perspectives?), the gap narrows to parity. W_1's proportional advantage persists through generation, not just retrieval.

Table 2: Generation evaluation (k=5): 5-judge majority vote, position-controlled blind pairwise vs. Top-k (N=156 queries). Fair% = W/(W+L). Win% = W/N. All approaches significant at $p<0.001$ (sign test) except Random. k=10 in Appendix G.1.3.

Approach	W/L/T	Fair%	Win%
Random	16/36/104	31%	10%
MMR*	39/8/109	83%	25%
DPP*	54/7/95	**89%**	35%
WassRank OT*	66/28/62	70%	**42%**
W_1-MMR*	44/10/102	81%	28%
W_1 Minimizer*	51/8/97	**86%**	33%

Metric isolation. On sparse pools, W_1-MMR appears to lose to KL(JS)-MMR (Appendix C.1), but the comparison is confounded: KL(JS)-MMR selects 1.7–2× more entity-matched documents than W_1-MMR on Yelp and OpinRank. Once we control for EM%, the Minimizer comparison flips: W_1 wins on both domains (Appendix C.1), and a single-judge generation evaluation confirms the same direction, with W_1 Minimizer leading KL(JS) Minimizer by 7 pp Fair% cross-domain (Appendix G.2).

Two other patterns are worth noting. First, semantic-diversity baselines fade as k grows: MMR and OpinionMMR Fair% collapse to 29–33% at k=20, while distributional methods keep gaining (WassRank OT: 70%→73%→92% across $k\in\{5, 10, 20\}$; Appendix G.1.3). Second, DPP and W_1 optimize different targets — bin diversity vs. bin ratios — which happen to align at k=5 with 2–3 active bins but diverge at k=20. On OpinRank, near-uniform P_{pop} compresses ordinal distances, so the KL/W_1 gap narrows; the ordinal advantage is largest where distributions are skewed (Yelp, Seller Forums).

4.4 Deployment Characterization

Which algorithm to pick comes down to one number: entity match percentage (Table 3). Dense pools, where all retrieved documents discuss the query entity, are easy. When EM≥85%, W_1 Minimizer delivers 69–81% reduction, statistically indistinguishable from oracle stratified sampling. Entity-Gated re-retrieval pushes Yelp further (W_1: 2.76→1.52, EM%: 93.8→99.2); on OpinRank both expansion strategies detect no deficit and self-bypass. Sparse entity pools are a different story. On Seller Forums (Top-k baseline EM∼40%; W_1 Minimizer reaches ∼51% post-reranking), W_1-MMR scores all 200 candidates regardless of entity match,

Table 3: Deployment decision matrix. Pool density determines algorithm choice; all methods SLA-compliant (N=200, k=20, m=7 bins). $|C_e|$ = entity-matched candidates. Noise tolerance details in Appendix F.1.

Pool Condition	Method	ms	Complexity
Dense (EM≥85%)	W_1 Minimizer	9	$O(\|C_e\| \cdot k \cdot m)$
Sparse (baseline EM∼40%)	W_1-MMR	154	$O(N \cdot k \cdot m)$
Variable / unknown	EG + W_1 Min	13	$O(\|C_e\| \cdot k \cdot m)$

reaching 43% reduction while maintaining 81% generation propagation (Table 2). This outperforms WassRank OT (70%) despite comparable retrieval-side W_1, making W_1-MMR the preferred hybrid for sparse and variable pools. Re-ranking latency stays under 310 ms across the board; end-to-end depends on retrieval infrastructure (Appendix B.5). For dense pools N=100 suffices; W_1-MMR benefits from larger N on sparse pools (Appendix F.3).

An exact P_{pop} turns out to be unnecessary: fifty entity-matched documents preserve 74–98% of oracle improvement; smoothing with a domain prior (τ=50) matches the oracle's harm rate. Cold-start entities fall back to this prior while still beating Top-k (Appendix F.4). The pipeline runs unchanged across managed cloud vector stores and local FAISS indices.

4.5 Independent-Target Validation

A natural concern is circularity: our LLM-derived sentiment-intensity (SI) labels define both the population target P_{pop} used at inference and the W_1 metric applied to the retrieved evidence. If the labeler had systematic biases, calibrating and measuring against it could inflate the numbers. To rule this out, we cross-validate against Yelp's native 1–5 star ratings on all 13,533 reviews. Stars are ordinal, human-authored at review time, and participate in *no part* of the WARP pipeline — neither retrieval, nor re-ranking, nor the P_{pop} target. Mapped to the same 7-bin SI scale, star-derived and LLM-derived labels correlate at Spearman ρ=0.881 ($p<10^{-300}$) and Pearson r=0.908, with 81.3% same-valence agreement and MAE 7.2 SI units — less than one bin (Appendix E).

Correlation on individual labels is necessary but not sufficient; a labeler could still consistently mis-estimate the entity-level P_{pop}. We therefore recompute P_{pop} from star ratings alone for each of the 10 Yelp entities and evaluate WARP's retrievals against this fully independent target. Mean W_1 gap between star-derived and LLM-derived P_{pop}: 2.88 (Appendix Table 16). Triangle inequality then gives $W_1(\text{WARP}, \text{star-}P_{\text{pop}}) \leq 5.64$ and $W_1(\text{Top-}k, \text{star-}P_{\text{pop}}) \geq 10.72$, i.e., a worst-case reduction of at least 47.4% *against a target our labels never touched*. This complements the mislabel sensitivity study (Appendix F), where W_1 Minimizer never crosses the Top-k baseline on Yelp even at 50% label corruption, along with the FDR-corrected significance testing and entity-clustered bootstrap in Appendix D. The gains are not an artifact of the labeler or of any single evaluation protocol.

5 Conclusion

As enterprise document corpora grow, RAG systems have become the default interface between users and organizational knowledge, returning relevant evidence in sub-second re-ranking latencies. Yet when the underlying corpus is diverse and highly opinionated, relevance-ranked retrieval selects only for topical similarity; the retrieved set may not reflect population opinions proportionally, producing summaries that appear grounded but misrepresent the balance of views.

WARP closes this gap by re-ranking retrieved evidence against the target population distribution, requiring no model fine-tuning, no retriever changes, and no additional inference calls. The result is at least a 43% reduction in distributional distance to the population target (W_1), all within a 310 ms latency envelope suitable for production deployment.

Limitations

Offline evaluation under production constraints. All reported results come from offline experiments, engineered to satisfy the sub-310 ms *re-ranking* budget typical of a large-scale e-commerce RAG setting (end-to-end depends on retrieval infrastructure; Appendix B.5). We therefore characterize retrieval and generation-side fidelity, not live user outcomes: whether proportionally faithful summaries measurably shift user trust, engagement, or decision quality under live traffic is untested, and offline W_1 reductions need not map one-to-one onto user-perceived faithfulness. A controlled online A/B test is the natural next step.

Corpus bias vs. retrieval bias. P_{pop} is *corpus-observed*, not a true population target: review corpora are self-selected, motivated writers with strong positive or negative experiences are over-represented, and some segments never write at all.

WARP removes the *retrieval-induced* bias — Top-k layers cosine-similarity skew on top of the underlying corpus bias — but does not remove the corpus bias itself. Faithful matching over a biased sample can lend false authority: the summary reflects the writers' distribution, not the underlying population's. Because P_{pop} is an *input* to the re-ranker rather than something it learns, external signals (native star ratings, demographic priors, survey-calibrated distributions) can be plugged in directly to reweight the target — a natural extension requiring no architectural change.

Pre-computed labels required. Per-document sentiment-intensity annotations are needed at indexing time. A single LLM pass suffices, and 50–100 labels per entity are enough - but the cost–quality tradeoff of labeling strategies is not evaluated.

Scale and Temporal Drift. 156 queries across 26 entities; paired Wilcoxon provides adequate power, but scaling beyond 14K documents is untested. P_{pop} degrades gracefully under perturbation (72% better than Top-k at ε=0.2) but temporal drift is not addressed.

Single ordinal axis; multi-issue opinions. WARP operates on a single ordinal axis (sentiment intensity). Multi-issue opinion spaces — e.g., a review that praises price but criticizes support, or patient-experience, employee-engagement, and multi-issue polling settings where several stance dimensions matter jointly — would require multi-marginal optimal transport, a natural extension we do not evaluate here.

Domain scope and generation. All domains are product/service reviews requiring entity-anchored opinions and an ordinal scale. Generation significance is driven by Yelp (97% decided) and OpinRank (94%); Seller Forums differentiates weakly (54%) due to sparse pools. A prompt ablation (Appendix G.1.4) shows reported gaps are criterion-dependent.

Ethics Statement

Distributional fidelity proportionally surfaces minority views, including potentially harmful ones. A content-safety filter should gate re-ranker output: excluded opinions are removed from both P_{pop} and the candidate pool *before* calibration. Because re-ranking is post-retrieval, it is compatible with any upstream safety filter without modification.

Use of AI Writing Assistance

The research (design, experiments, analysis, and writing) is the authors' own work. We used Claude (Anthropic) as a proofreader and sounding board: it flagged unclear phrasing, suggested structural edits, and helped scaffold parts of the experimental code, all of which we reviewed and revised ourselves. Figure 1 was generated with ChatGPT (OpenAI) to give readers a quick visual overview of the pipeline; we verified its accuracy. No AI system produced research claims or final prose.

References

Aditya Agrawal, Alwarappan Nakkiran, Darshan Fofadiya, Alex Karlsson, Harsha Aduri, and Aman Singh Thakur. 2026. Retrieval-augmented generation must move beyond factual grounding to represent diverse opinions. *Preprint*, arXiv:2604.12138.

Elif Aktolga and James Allan. 2013. Sentiment diversification with different biases. In *Proceedings of ACM SIGIR*, pages 593–602.

Amazon AGI. 2025. The Amazon Nova family of models: Technical report and model card. *Preprint*, arXiv:2506.12103.

Amazon Web Services. 2024a. Amazon Titan text embeddings v2. `https://docs.aws.amazon.com/ai/responsible-ai/titan-text-embeddings/overview.html`. Accessed: 2026-05.

Amazon Web Services. 2024b. Knowledge bases for Amazon Bedrock. `https://docs.aws.amazon.com/bedrock/latest/userguide/knowledge-base.html`. Accessed: 2026-05.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Anthropic. 2025. System card: Claude Opus 4 & Claude Sonnet 4. `https://www-cdn.anthropic.com/6be99a52cb68eb70eb9572b4cafad13df32ed995.pdf`. Accessed: 2026-05.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of ACM SIGIR*, pages 335–336.

Sebastian Claici, Aude Genevay, and Justin Solomon. 2018. Wasserstein measure coresets. *Preprint*, arXiv:1805.07412.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

David F. Crouse. 2016. On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696.

Van Dang and W. Bruce Croft. 2012. Diversity by proportionality: An election-based approach to search result diversification. In *Proceedings of ACM SIGIR*, pages 65–74.

DeepSeek-AI. 2025. DeepSeek-V3.2: Pushing the frontier of open large language models. *Preprint*, arXiv:2512.02556.

Kavita Ganesan and ChengXiang Zhai. 2012. Opinion-based entity ranking. *Information Retrieval*, 15(2):116–150.

Han Jiang, Rui Wang, Zhihua Wei, Yu Li, and Xinpeng Wang. 2023. Large-scale and multi-perspective opinion summarization with diverse review subsets. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5641–5656, Singapore. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Alex Kulesza and Ben Taskar. 2012. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286.

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *Proceedings of ICML*, pages 957–966.

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. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Qiheng Lu and Nicholas D. Sidiropoulos. 2026. Principled and scalable diversity-aware retrieval via cardinality-constrained binary quadratic programming. *Preprint*, arXiv:2604.02554. Concurrent work.

Meta. 2024. Llama 3.3 model card. `https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_3/`. Accessed: 2026-05.

Mistral AI. 2025. Mistral large 3. `https://docs.mistral.ai/models/model-cards/mistral-large-3-25-12`. Accessed: 2026-05.

Mir Tafseer Nayeem and Davood Rafiei. 2025. OpinioRAG: Towards generating user-centric opinion highlights from large-scale online reviews. In *Second Conference on Language Modeling*.

Vu Nguyen and Andrey Kan. 2026. MUSS: Multilevel subset selection for relevance and diversity. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI)*.

Alex Oesterling, Claudio Mayrink Verdun, Carol Xuan Long, Alexander Glynn, Lucas Monteiro Paes, Sajani Vithana, Martina Cardone, and Flavio P. Calmon. 2024. Multi-group proportional representation in retrieval. In *Advances in Neural Information Processing Systems*, volume 37, pages 114601–114655. Curran Associates, Inc.

Chao Peng, Bin Wang, Zhilei Long, and Jinfang Sheng. 2025. AdaGReS: Adaptive greedy context selection via redundancy-aware scoring for token-budgeted RAG. *Preprint*, arXiv:2512.25052.

Gabriel Peyré and Marco Cuturi. 2019. Computational optimal transport: With applications to data science. *Foundations and Trends in Machine Learning*, 11(5–6):355–607.

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

Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis. 2010. Exploiting query reformulations for web search result diversification. In *Proceedings of WWW*, pages 881–890.

Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. 2025. Judging the judges: A systematic study of position bias in LLM-as-a-judge. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 292–314, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In *Proceedings of ACM SIGKDD*, pages 2219–2228.

Harald Steck. 2018. Calibrated recommendations. In *Proceedings of ACM RecSys*, pages 154–162.

Peggy Tang, Kun Hu, Rui Yan, Lei Zhang, Junbin Gao, and Zhiyong Wang. 2022. OTExtSum: Extractive text summarisation with optimal transport. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1128–1141, Seattle, United States. Association for Computational Linguistics.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM^2)*, pages 404–430, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of NeurIPS*.

Haolun Wu, Yansen Zhang, Chen Ma, Fuyuan Lyu, Bowei He, Bhaskar Mitra, and Xue Liu. 2024. Result diversification in search and recommendation: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 36(10):5354–5373.

Yelp. 2024. Yelp open dataset. `https://www.yelp.com/dataset`. Accessed: 2026-05.

Hai-Tao Yu, Adam Jatowt, Hideo Joho, Joemon M. Jose, Xiao Yang, and Long Chen. 2019. WassRank: Listwise document ranking using optimal transport theory. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 24–32. Association for Computing Machinery.

Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. 2017. FA*IR: A fair top-k ranking algorithm. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1569–1578. ACM.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36.

A Experimental Setup

A.1 Dataset Details

Opinion Scale. Sentiment-Intensity (SI) is mapped to a discrete ordinal scale, shown in Table 4.

Table 4: Sentiment-Intensity (SI) mapping to the 7-bin ordinal scale. *Mixed* labels (praise-and-criticism co-occurring in one review) collapse to the neutral bin; they account for 4–8% of extractions across domains and W_1 Minimizer never crosses the Top-k baseline on Yelp even at 50% label corruption (§F.1), so this pooling is bounded in impact. Multi-issue axes require multi-marginal transport — outside our current scope (§Limitations).

Sentiment	Intensity	SI Score
Positive	High / Med / Low	+30 / +20 / +10
Neutral	Any	0
Mixed	—	0
Negative	High / Med / Low	-30 / -20 / -10

Entity Selection. Three stages. First, LLM-based entity extraction with domain-specific seed lists. Then diversity filtering: Shannon entropy $H \geq 0.6$, minority sentiment $\geq 10\%$, mention count ≥ 100. Finally, top-n selection ranked by entropy descending.

A.2 Query Templates

Table 6 lists the exact query templates used to generate evaluation questions. Each entity is instantiated into all 6 templates, yielding 156 total queries (36 Seller Forums + 60 Yelp + 60 OpinRank).

Entities. **Seller Forums** (6): A+ Content, Account Health Rating, Brand Stores, Community Management, Coupons, Seller Education. **Yelp Hotels** (10): Booking Experience, Breakfast, Breakfast Quality, Business Center, Casino, Loyalty Program, Nightly Rate, Pool, Room Quality, Shuttle Service. **OpinRank Cars** (10): Oil Change Interval, Dealer Experience, Vehicle Size, Sales Experience, Fuel Tank Size, Resale Value, City MPG, Remote Start, Climate Control, Ground Clearance.

A.3 Prompt Library

All prompts used in the WARP pipeline. Template variables shown in `{braces}`.

A.3.1 Opinion Extraction

Prompt A.3.1.1: Registry-Guided Extraction (Open-Domain)

A single LLM call per document extracts entities and sentiment-intensity labels using a domain-specific tiered seed list while remaining open to discovering new entities.

Table 5: Dataset summary across three evaluation domains.

Dataset	Domain	Source	Size	Entities	Queries	Search	Embedding	Chunking	Labeling
Seller Forums	E-comm. seller	Public	∼8K	6	36	Hybrid	Titan V2 (Amazon Web Services, 2024a) (1024d)	300 tok / 20% overlap	LLM-enriched metadata
Yelp Hotels	Hospitality	Yelp Open	14K	10	60	Semantic	MiniLM-L6 (384d)	Whole review	LLM-extracted (Claude)
OpinRank Cars	Automotive	OpinRank	13K	10	60	Semantic	MiniLM-L6 (384d)	Whole review	LLM-extracted

Table 6: Query templates per domain. Each template is instantiated with every selected entity (6 for Seller Forums, 10 for Yelp/OpinRank), producing 6 questions per entity (2 breadth, 2 polar, 2 segment).

Type	Idx	Seller Forums	Yelp Hotels	OpinRank Cars
breadth	0	What do sellers think about {entity}?	What do guests think about {entity}?	What do owners think about {entity}?
breadth	1	Summarize seller opinions on {entity}.	Summarize guest opinions on {entity}.	Summarize owner opinions on {entity}.
polar	0	What are the biggest complaints about {entity}?	What are the biggest complaints about {entity}?	What are the biggest complaints about {entity}?
polar	1	What positive experiences have sellers had with {entity}?	What positive experiences have guests had with {entity}?	What do owners praise most about {entity}?
segment	0	How do small sellers vs large sellers feel about {entity}?	How do business travelers vs leisure travelers feel about {entity}?	How do commuters vs enthusiasts feel about {entity}?
segment	1	Do new sellers and experienced sellers differ in their views of {entity}?	Do solo guests and families differ in their views of {entity}?	Do new buyers and long-term owners differ in their views of {entity}?

System: You extract structured opinion data from reviews. Output ONLY valid JSON arrays. No markdown, no explanation.

User: Extract entity-level opinions from this review.

`[{registry_block} -- domain-specific tiered entity list injected here]`

Instructions:
1. Identify ALL entities/aspects the reviewer discusses.
2. If an entity matches the registry above, use the EXACT registry name. Prefer the most specific tier (Tier 3 > Tier 2 > Tier 1).
3. If the reviewer discusses something NOT in the registry, create a short descriptive entity name (open discovery).
4. For EACH entity mentioned, output:
- entity: the entity name (registry canonical or new)
- sentiment: "positive", "negative", "neutral", or "mixed"
- intensity: "high", "medium", or "low"
- evidence: exact quote from the review supporting this opinion (max 120 chars)

Only include entities explicitly discussed. Return `[]` if none found.
Output ONLY a JSON array.

Review:
`{review_text}`

For closed-domain settings (e.g., Yelp Hotels), the extraction prompt uses a fixed aspect list with keyword mappings instead of the tiered registry, but the output schema is identical.

A.3.2 Answer Generation

Prompt A.3.2.1: Opinion Summary Generator

Instructs the LLM to synthesize a proportional opinion summary from retrieved documents.

System: You are summarizing community opinions from review data. Use ONLY the provided documents. Represent opinions proportionally — include minority views, not just the majority. Be specific and cite reviewer experiences where possible.

User: Question: `{question}`

Reviews:
`{docs}`

Summarize what the community thinks. Be thorough — cover the full range of opinions proportionally.
Do not add opinions beyond what's in the documents.

A.3.3 Pairwise Generation Evaluation (Judge)

Each answer pair is evaluated via position-controlled blind pairwise comparison (positions swapped in the second call for debiasing). The judge scores three dimensions; the OPINION FAIRNESS dimension is swapped between variants in our ablation (§G.1.4).

Prompt A.3.3.1: Pairwise Judge Template

Full evaluation template: blind comparison on opinion fairness, informativeness, and groundedness.

System: You are an expert evaluator comparing two answers that summarize community opinions. You do NOT know which retrieval method produced which answer. Judge strictly on the quality of the answers themselves.

User: You are evaluating two answers to the same question about community opinions. Both answers were generated from different sets of reviews retrieved for the same question.

Question: `{question}`

— Answer A —
`{answer_a}`

— Answer B —
`{answer_b}`

Compare the answers on these dimensions. For each, pick "A", "B", or "tie":

1. OPINION FAIRNESS: `[fairness variant inserted -- see Prompts A.3.3.2 and A.3.3.3 below]`

2. INFORMATIVENESS: Which answer is more helpful, specific, and actionable for someone trying to understand what the community thinks? Consider breadth of points covered, specificity, and usefulness.

3. GROUNDEDNESS: Which answer appears more grounded in actual community member experiences? Look for attribution to specific perspectives vs vague generalizations that could be hallucinated.

Respond ONLY with JSON:
`{"fairness": "A"|"B"|"tie", "informativeness": "A"|"B"|"tie", "groundedness": "A"|"B"|"tie", "reasoning": "<1-2 sentences>"}`

The baseline evaluation uses a generic fairness instruction ("which answer more proportionally represents the full range of community opinions, including minority views?"). Our ablation replaces it with the two variants below.

Prompt A.3.3.2: Fairness Variant A — Proportional Accuracy

Provides ground-truth P_{pop} to the judge; penalizes over-representing minority opinions.

OPINION FAIRNESS (Proportional Accuracy): The actual community opinion distribution for this topic is: `{p_pop_description}`. Which answer more accurately reflects these real proportions? The ideal answer should make the reader walk away with a correct sense of how common each view is: a dominant opinion should dominate the summary, a rare opinion should be mentioned but not over-emphasized. An answer that gives equal weight to a 10% minority and a 70% majority is MISLEADING, even if well-intentioned. Pick the answer whose emphasis better matches the actual distribution above.

Where `{p_pop_description}` is dynamically filled per entity from the ground-truth P_{pop}, e.g.: "approximately 65% positive (mostly satisfied), 25% negative (dissatisfied), 10% neutral."

Prompt A.3.3.3: Fairness Variant B — Viewpoint Coverage

Rewards distinct viewpoint coverage; favors surfacing rare perspectives even if disproportionate.

OPINION FAIRNESS (Viewpoint Coverage): Which answer covers MORE DISTINCT VIEWPOINTS from the community, giving voice to all perspectives including rare and minority ones? An answer that only represents the majority view — even if that majority is large — is LESS fair than one that surfaces unique perspectives readers might not otherwise encounter. The ideal answer ensures no viewpoint goes unheard, even if that means giving disproportionate space to rare opinions.

A.3.4 Retrieval Augmentation

Prompt A.3.4.1: Pole-Biased Multi-Query Templates

During pool expansion (Stage 1.5), three sentiment-pole queries recover documents that cosine retrieval buries.

Positive: What positive experiences have sellers had with `{entity}`?

Negative: What are the biggest complaints about `{entity}`?

Neutral: What factual information do sellers share about `{entity}`?

Evaluation queries (6 per entity) are generated from three templates—breadth ("What do `{persona}` think about `{entity}`?"), polar ("What are the biggest complaints..."), and segment ("How do `{segment_A}` vs `{segment_B}` feel...")—with domain-adapted persona and segment terms (e.g., "sellers" / "small sellers vs large sellers").

B Algorithm Details

Table 7 summarizes the key symbols used throughout.

Table 7: Notation reference.

Symbol	Meaning
d	A candidate document
e	Target entity
k	Output size (documents returned)
N	Candidate pool size
$SI(d)$	Sentiment-intensity label of d
P_{pop}	Ground-truth population distribution
P_{pool}	Pool-level empirical distribution
$W_1(P, Q)$	Wasserstein-1 distance between P and Q
λ	Relevance–calibration trade-off weight
δ	Deficit threshold for pool expansion
n_{exp}	Budget (docs per deficit pole)
$\text{rel}(d)$	Relevance score of document d
EM	Entity-matched subset of pool

B.1 Entity-Gated Re-retrieval

The two-pass entity-gated pool construction recovers entity-matched documents that pure relevance ranking buries.

Algorithm 2 Entity-Gated Re-retrieval

Require: Results R (by score), entity e, sizes n_1, n_2, threshold τ

$P_1 \leftarrow R[1 : n_1]$ (top-n_1 by relevance)
$C_2 \leftarrow \{d \in R \setminus P_1 : \text{entity}(d) = e \wedge \text{score}(d) \geq \tau\}$
Sort C_2 by $|SI(d)|$ descending
$P_2 \leftarrow C_2[1 : n_2]$
Pool $\leftarrow$ dedup$(P_1 \cup P_2)$, sorted by score
return Pool

B.2 Adaptive Distribution-Aware Pool Expansion

When entity-matched documents under-represent a sentiment pole relative to P_{pop}, pole-biased queries (§A.3.4) retrieve additional candidates before re-ranking. The cold-start path (EM $= \emptyset$) treats every pole with non-negligible mass as a deficit, ensuring new entities still receive balanced pools. In practice, one to three deficit poles trigger per entity; n_{exp} caps total expansion cost at a fixed multiple of k.

Algorithm 3 Adaptive Pool Expansion

Require: Pool C_0, entity e, P_{pop}, threshold δ, budget n_{exp}

EM $\leftarrow \{d \in C_0 : \text{entity}(d) = e\}$
if EM $= \emptyset$ **then**
  DeficitPoles $\leftarrow \{s : P_{\text{pop}}(s) > 0.01\}$
else
  $P_{\text{pool}}(s) \leftarrow |\{d \in \text{EM} : SI(d) = s\}|/|\text{EM}|$
  DeficitPoles $\leftarrow \{s : P_{\text{pool}}(s) < \delta \cdot P_{\text{pop}}(s)\}$
end if
if DeficitPoles $= \emptyset$ **then**
  return C_0
end if
Retrieve n_{exp} entity-matched docs per deficit pole
Pool $\leftarrow$ dedup$(C_0 \cup C_{\text{exp}})$
return Pool

B.3 W_1-MMR

The W_1-MMR variant blends relevance with marginal Wasserstein reduction, mirroring the MMR trade-off while targeting distributional calibration rather than pairwise dissimilarity.

Algorithm 4 W_1-MMR

Require: Pool C, scores rel$(\cdot)$, target P_{pop}, λ, size k

$S \leftarrow \emptyset$
while $|S| < k$ **do**
  $W_1^{\text{cur}} \leftarrow \begin{cases} 60 & \text{if } S = \emptyset \\ W_1(\text{EmpDist}(S), P_{\text{pop}}) & \text{else} \end{cases}$
  for each $d \in C \setminus S$ **do**
    $\Delta W_1 \leftarrow W_1^{\text{cur}} - W_1(\text{EmpDist}(S \cup \{d\}), P_{\text{pop}})$
    $\text{score}(d) \leftarrow \lambda \cdot \text{rel}(d) + (1-\lambda) \cdot \Delta W_1 / W_1^{\text{cur}}$
  end for
  $d^* \leftarrow \arg\max_{d \in C \setminus S} \text{score}(d)$
  $S \leftarrow S \cup \{d^*\}$
end while
return S

B.4 Label Acquisition

Per-document SI labels come from a single LLM pass at indexing time (<200 ms/doc). Labeling is cheap. The Seller Forums corpus (∼8K documents) labels in under 4 minutes with 150 concurrent workers. P_{pop} is just a count aggregation per entity ($O(n)$, milliseconds); incremental updates on ingestion avoid recomputation entirely.

B.5 End-to-End Latency Breakdown

Hardware. Intel Xeon Platinum 8175M @ 2.50 GHz, 8 cores (single-threaded execution), no GPU.

WARP's re-ranking step runs entirely in NumPy on the retrieved candidate distribution — no additional model calls. End-to-end latency therefore decomposes into three components (Table 8):

Re-ranking latency is dominated by CDF differencing over m=7 bins and is backend-independent. Retrieval and re-retrieval dominate managed-endpoint deployments; for local FAISS the whole pipeline fits inside a 500 ms sub-second SLA.

The W_1 Minimizer's near-zero p50 on Seller Forums (0.1 ms) reflects the sparse entity-filtered pool: few candidates match the queried entity, so the greedy loop terminates almost immediately. p99 stays under 310 ms across all WARP variants on both domains, and end-to-end with local FAISS retrieval fits in <365 ms — comfortably inside a sub-second SLA. Managed vector stores add ∼1.5 s of retrieval latency that WARP does not remove.

Table 8: Per-component latency breakdown. Re-ranking is hardware-only (no model inference); retrieval and re-retrieval depend on the vector-store backend. p99 for re-ranking; typical/median for the rest.

Component	Local FAISS	Managed Bedrock Knowledge Base
Retrieval	<50 ms	∼1.5 s
Re-retrieval†	<5 ms	∼100 ms
Re-ranking (p99)	<310 ms	<310 ms
End-to-end	**<365 ms**	**∼1.9 s**

† Only fires when pool expansion (Entity-Gated / Adaptive) detects a deficit.

Table 9: Per-query re-ranking latency percentiles (N=96 queries: 36 Seller Forums + 60 Yelp Hotels, single-threaded on the hardware above). p99 drives production SLAs; max reported for tail auditing. All WARP variants stay under 310 ms at p99.

Domain	Method	Queries	p50 (ms)	p95 (ms)	p99 (ms)	Max (ms)
Seller Forums	W_1 Minimizer	36	**0.1**	**38.7**	**45.5**	**48.0**
Seller Forums	W_1-MMR	36	156.8	163.1	166.2	167.4
Seller Forums	EG + W_1 Min	36	0.6	211.0	216.2	218.8
Seller Forums	DPP	36	47.8	49.1	51.9	53.0
Yelp Hotels	W_1 Minimizer	60	**65.0**	270.8	289.9	295.3
Yelp Hotels	W_1-MMR	60	81.6	284.7	301.6	303.5
Yelp Hotels	EG + W_1 Min	60	145.7	302.0	307.6	312.5
Yelp Hotels	DPP	60	103.5	**127.4**	**129.3**	**131.3**

C Retrieval Ablations

We ablate key components of the retrieval pipeline: distance metric choice, entity filtering, pool expansion, convergence, per-entity variation, and sensitivity to hyperparameters.

C.1 JS vs. W_1: Metric Comparison and Entity-Match Confound

This section unpacks the JS-vs-W_1 comparison summarized in Section 4.2. We rerun the four calibrated methods (KL(JS) Minimizer, W_1 Minimizer, KL(JS)-MMR, W_1-MMR) on Yelp and OpinRank at k=20, holding the greedy structure and entity filtering fixed so only the distance metric varies within each row-pair.

Table 10 shows KL(JS)-MMR beating W_1-MMR on Yelp and OpinRank W_1. On its face this reverses the ordinal-metric advantage the Minimizer comparison establishes. It doesn't: the two hybrids retrieve very different candidate pools, and the metric is being scored on top of that difference.

In the top block (hybrids, uncontrolled EM), KL(JS)-MMR retrieves nearly twice as many entity-matched documents as W_1-MMR (38.2% vs. 19.4% on Yelp; 36.1% vs. 21.8% on OpinRank), and its lower W_1 tracks that EM gap rather than the choice of ordinal-vs-categorical ground cost. In the bottom block (Minimizers, controlled EM), both algorithms filter to the same entity-matched pool (EM=93.8% Yelp, 85.8% OpinRank), and W_1 wins on both domains — 2.76 vs. 3.00 on Yelp, 4.14 vs. 4.54 on OpinRank. The controlled comparison is where the ordinal metric earns its billing; the uncontrolled hybrid comparison is a retrieval-pool artifact.

Table 10: Entity-match confound in hybrid variants. KL(JS)-MMR retrieves ∼1.7–2× more entity-matched documents than W_1-MMR on both Yelp and OpinRank. In the controlled Minimizer comparison (same entity filter, EM equalized) W_1 wins on both domains — the KL(JS)-MMR W_1 advantage in the uncontrolled comparison traces to the EM gap, not to metric superiority.

	Yelp		OpinRank	
Method	W_1	EM%	W_1	EM%
KL (JS)-MMR	**2.51**	38.2%	**3.32**	36.1%
W_1-MMR	4.02	19.4%	3.41	21.8%
KL (JS) Minimizer	3.00	93.8%	4.54	85.8%
W_1 Minimizer	**2.76**	93.8%	**4.14**	85.8%

C.2 Entity-Filtered Control (Yelp)

Table 11: Entity-filter control experiment (Yelp Hotels, k=20). Entity filtering alone reduces W_1 by 45.7%; W_1 Minimizer achieves 62.6% additional reduction beyond the entity-filter control.

Approach	EM%	W_1	Δ vs Top-K	Δ vs EF+Top-K
Top K (no filter)	40.1	13.60	—	—
EntityFilter + Top-K	100.0	7.38	−45.7%	—
EntityFilter + MMR	100.0	6.99	−48.6%	−5.3%
EntityFilter + DPP	100.0	5.64	−58.5%	−23.6%
W_1 Minimizer	93.8	2.76	−79.7%	−62.6%
W_1-MMR	19.4	4.02	−70.4%	−45.5%
EG + W_1 Min	99.2	1.52	−88.8%	−79.4%
AE + W_1 Min	95.9	1.66	−87.8%	−77.5%

On sparse Seller Forums (Top-k baseline EM∼40%; entity-filter ceiling ∼51%), entity fil-

tering provides only -11.6% W_1 reduction (vs. -45.7% on Yelp, where the entity filter reaches 100% EM), confirming that W_1-MMR's full-pool scoring drives gains in entity-sparse domains (-37.7% beyond the entity-filter baseline).

C.3 Adaptive-k Convergence

W_1 Minimizer hits diminishing returns at k^*=3 (Seller Forums), k^*=4 (Yelp), k^*=5 (OpinRank) (Figure 3). An adaptive-k strategy could therefore feed far fewer documents to the LLM without losing distributional fidelity. This also implies robustness to imprecise P_{pop} estimates.

C.4 W_2 vs. W_1 Comparison

A natural concern is whether optimizing W_1 (linear ground cost) sacrifices performance under higher-order transport metrics. We compute W_2 (quadratic ground cost) for all methods on Yelp Hotels (k=20). Across the distributions produced by these methods, we observe that the method ranking under W_1 is preserved under W_2, with a linear relationship (slope $= 1.19$, $R^2 > 0.99$). W_1 is chosen for its $O(m)$ closed-form computation.

C.5 Per-Entity Breakdowns

Aggregate results in Table 1 mask substantial per-entity variation. The gains concentrate on entities whose baseline distributions are most skewed, precisely where minority-opinion misrepresentation does the most damage. We break out per-entity W_1 for all three domains below.

C.5.1 Yelp Hotels (k=20)

W_1 Minimizer lands below 2.0 on 7 of 10 entities. That is near-perfect calibration. Three outliers remain above 2.0: Business Center (9.76), Shuttle Service (5.78), and Loyalty Program (4.07), all corresponding to entities with extreme distributional skew and thin pool diversity. Even greedy optimization cannot fully close the gap within k=20 selections under those conditions.

Table 12: Per-entity breakdown (Yelp Hotels, 10 aspects). W_1 Minimizer achieves <2.0 on 7/10 entities.

Entity	Top-k W_1	W_1 Min	W_1-MMR	WassRank OT
Booking Experience	18.30	1.95	3.78	1.58
Breakfast	7.55	1.48	2.87	1.42
Breakfast Quality	5.78	0.87	3.79	0.65
Business Center	24.51	9.76	6.72	9.26
Casino	14.79	0.86	4.07	0.62
Loyalty Program	16.31	4.07	4.49	2.95
Nightly Rate	7.49	0.82	3.45	0.66
Pool	10.79	0.73	2.68	0.46
Room Quality	8.55	1.24	2.67	2.18
Shuttle Service	21.95	5.78	5.70	5.38

C.5.2 OpinRank Cars (k=20, top 10 by entropy)

OpinRank entities vary widely in mention count (39–1,465) and minority fraction (19–45%). High-count entities with moderate minority shares calibrate well: City MPG, for instance, has 1,465 mentions at 25% minority and reaches W_1=0.38. Hard cases persist. Vehicle Size has only 42 mentions; Dealer Experience starts from a W_1=23.07 baseline. Both resist full correction.

D Statistical Validity

Two orthogonal concerns for the significance testing: multiple comparisons across many method $\times$ domain combinations, and non-independent samples within a domain (queries drawn from the same entity share an underlying opinion distribution). We address both.

D.1 Benjamini–Hochberg Correction

We ran 57 paired Wilcoxon tests spanning three domains, multiple re-ranking variants (baselines and W_1 family), and all evaluated k values. Benjamini–Hochberg FDR correction at α=0.05 leaves 39/57 tests surviving. Table 14 shows the 14 W_1-family tests: every one survives on every domain. The 18 non-surviving tests come from baseline re-ranking methods outside our contribution set.

Table 14: Benjamini–Hochberg FDR correction (α=0.05) on paired-Wilcoxon p-values for the W_1 family. Every W_1-family algorithm survives on every domain (14/14). Reduction is per-query paired reduction relative to Top-k; aggregate reductions in Table 1 weight queries differently. $^{***}p_{\text{adj}}<10^{-3}$, $^{**}p_{\text{adj}}<10^{-2}$, $^{*}p_{\text{adj}}<0.05$.

Domain	Approach	N	Reduction	p_{adj}
Yelp	W_1 Minimizer***	57	+81.4%	<0.000001
Yelp	W_1-MMR***	57	+70.2%	<0.000001
Yelp	EG + W_1 Min***	57	+89.8%	<0.000001
Yelp	AE + W_1 Min***	57	+89.1%	<0.000001
Yelp	WassRank OT***	57	+83.3%	<0.000001
OpinRank	W_1 Minimizer***	54	+72.4%	<0.000001
OpinRank	W_1-MMR***	54	+76.7%	<0.000001
OpinRank	EG + W_1 Min***	54	+72.4%	<0.000001
OpinRank	AE + W_1 Min***	54	+72.4%	<0.000001
OpinRank	WassRank OT***	54	+74.7%	<0.000001
Seller Forums	W_1 Minimizer***	24	+59.1%	0.000427
Seller Forums	W_1-MMR**	24	+20.4%	0.004552
Seller Forums	EG + W_1 Min*	24	+53.7%	0.049
Seller Forums	AE + W_1 Min*	24	+53.7%	0.049

D.2 Entity-Clustered Bootstrap

Paired-Wilcoxon assumes independent paired observations. Queries drawn from the same entity are not independent, since they share the same underlying opinion distribution. We therefore resample at

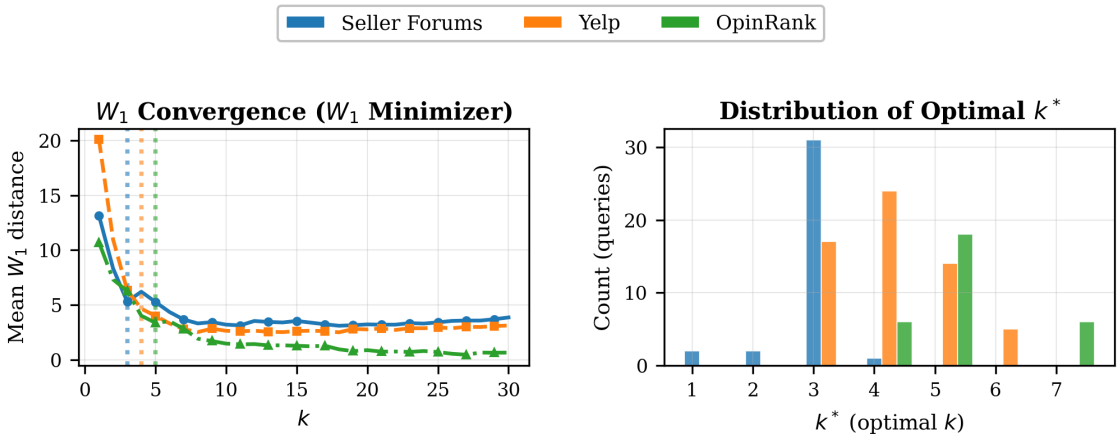

Figure 3: W_1 Minimizer convergence vs. output size k. Median convergence point k^* (where $\Delta W_1 < 0.5$ per additional document): Seller Forums k^*=3, Yelp k^*=4, OpinRank k^*=5.

Table 13: Per-entity breakdown (OpinRank, LLM-extracted entities). Count = opinion mentions, H = Shannon entropy, Min% = minority sentiment fraction.

Entity	Count	H	Min%	Top-k	W_1 Min	W_1-MMR	WassRank OT
Oil Change Interval	144	1.49	21%	3.96	1.08	2.95	0.78
Dealer Experience	425	1.47	30%	23.07	11.71	3.40	10.28
Vehicle Size	42	1.37	19%	0.00	8.31	6.19	8.31
Sales Experience	776	1.33	28%	15.66	3.01	3.19	2.57
Fuel Tank Size	269	1.32	36%	13.67	3.06	3.25	3.05
Resale Value	427	1.30	39%	12.19	2.24	2.40	2.33
City MPG	1465	1.29	25%	9.12	0.38	2.78	0.38
Remote Start	39	1.25	41%	8.80	4.52	3.54	4.52
Climate Control	727	1.24	45%	9.95	0.95	2.43	0.94
Ground Clearance	201	1.22	33%	23.55	6.12	4.00	5.50

the entity level: 10,000 bootstrap iterations, drawing entities with replacement and recomputing the mean per-query W_1 improvement. All 95% CIs exclude zero for every W_1 method on every domain (Table 15).

Table 15: Entity-clustered bootstrap (10,000 resamples). Mean improvement in W_1 relative to Top-k; 95% CIs computed by resampling entities with replacement. All CIs exclude zero.

Dataset	Method	#Entities	Mean ΔW_1	95% CI
Yelp	W_1 Minimizer	10	10.85	[8.26, 13.37]
Yelp	EG + W_1 Min	10	12.09	[8.72, 15.37]
Yelp	W_1-MMR	10	9.58	[6.40, 12.80]
OpinRank	W_1 Minimizer	9	9.65	[6.92, 12.34]
OpinRank	W_1-MMR	9	10.22	[6.48, 14.17]
Seller Forums	W_1 Minimizer	4	5.32	[0.67, 10.38]
Seller Forums	W_1-MMR	4	1.84	[0.49, 3.25]

Seller Forums' wider CIs (e.g., W_1 Minimizer [0.67, 10.38]) reflect its small entity count (N=4), not a qualitatively different effect — the lower bound still exceeds zero. The aggregate signal is cross-domain consistency across 10 Yelp, 9 OpinRank, and 4 Seller Forums entities.

E Independent-Target Validation

Our LLM-derived sentiment-intensity labels define both the population target P_{pop} used at inference and the W_1 evaluation metric applied to the retrieved evidence. If the labeler had systematic biases, calibrating and measuring against it could inflate the numbers. We cross-validate against Yelp's native 1–5 star ratings, which are ordinal, human-authored at review time, and participate in *no part* of the WARP pipeline — neither retrieval, nor reranking, nor the P_{pop} used at inference. Section 4.5 in the main body summarizes; this appendix reports the underlying numbers.

A high correlation on individual labels is a necessary but not sufficient condition: the entity-level P_{pop} is an aggregate, and a labeler could still consistently mis-estimate it. We therefore recompute P_{pop} from star ratings alone for each of the 10 Yelp entities and quantify the divergence from the LLM-derived P_{pop} actually used at inference (Table 16). Mean W_1 divergence: 2.88. Under a triangle-inequality bound, WARP's retrievals stay within W_1=5.64 of the star-derived target while Top-k is at least W_1=10.72 away — a worst-case reduction of at least 47.4% against a target the labeler never touched.

Table 16: Star-derived vs. LLM-derived P_{pop} per Yelp entity. Mean W_1 gap: 2.88. Bounds: $W_1(\text{WARP}, \text{star-}P_{\text{pop}}) \leq 5.64$, $W_1(\text{Top-}k, \text{star-}P_{\text{pop}}) \geq 10.72$; worst-case reduction $\geq 47.4\%$.

Entity	N **reviews**	$W_1(\text{star-}P_{\text{pop}}, \text{LLM-}P_{\text{pop}})$
Booking Experience	1,372	3.66
Breakfast	788	1.44
Breakfast Quality	637	4.15
Business Center	142	3.23
Casino	208	1.36
Loyalty Program	607	1.57
Nightly Rate	2,481	4.60
Pool	1,318	2.10
Room Quality	2,557	1.78
Shuttle Service	280	4.90
Mean	—	**2.88**

F Robustness & Sensitivity

F.1 Noise Tolerance

Two studies stress-test the system under realistic noise. First, Dirichlet perturbation of P_{pop} simulates inaccurate population estimates. Second, mislabel sensitivity corrupts both document SI labels and P_{pop} at once. These jointly reveal how much noise each algorithm can absorb, and where deployment breaks down.

F.1.1 Dirichlet Perturbation of P_{pop}

We perturb P_{pop} with Dirichlet noise at $\varepsilon \in \{0, 0.1, 0.2, 0.5\}$ (N=30 samples per entity) to simulate estimation error in the population distribution (Figure 4):

On Yelp, which has a dense candidate pool, the W_1 Minimizer at ε=0.2 degrades by 31%, but it still comes in 72% ahead of Top-k. W_1-MMR barely moves under the same perturbation (<1% degradation). On Amazon Seller Forums the picture is similar in shape: all methods degrade gracefully, and W_1 Minimizer at ε=0.2 stays 56% above Top-k. OpinRank is the most sensitive of the three because the unperturbed baseline is already tight (W_1 Min sits at 0.87 at ε=0), so relative degradation looks larger, but even there ε=0.2 holds 45% better than Top-k. Across all three domains, W_1-MMR degrades the least, which we attribute to its hybrid objective partially shielding it from target noise.

F.1.2 Mislabel Sensitivity (SI Label Error)

A harder question: at what SI label error rate does the W_1 advantage drop below significance? We corrupt candidate document labels AND P_{pop} simultaneously, the realistic scenario where labeling errors propagate into the population estimate. Error rates $\in \{0, 0.05, 0.10, 0.20, 0.30, 0.50\}$, N=20 repetitions per setting (Figure 5). "Crossover" marks the error rate at which a method's W_1 exceeds the Top-k baseline:

Dense pools are forgiving. W_1 Minimizer never crosses the Top-k baseline on Yelp, even at 50% label error. Sparse pools need more care; use W_1-MMR regardless of label quality, as its hybrid objective guarantees a floor.

F.2 Noise Control

We need to rule out a trivial explanation: maybe any score perturbation induces diversity that looks like opinion-aware signal. So we add 5% Gaussian noise to retrieval scores and re-rank:

Table 17: Noise control: 5% Gaussian perturbation of retrieval scores yields negligible W_1 improvement (2–5%, $p > 0.05$), confirming that our 43–89% distributional improvements (W_1-MMR on sparse pools; W_1 Minimizer / EG+W_1 Min on dense) represent genuine distributional optimization.

Method	**Seller** W_1	**Yelp** W_1	**OpinRank** W_1
Top-k (baseline)	10.33	13.60	13.33
Top-k + 5% Noise	10.07	13.00	12.97
Reduction	*2.5%*	*4.4%*	*2.7%*
Significance	*ns ($p > 0.05$, all domains)*		
W_1 Minimizer	8.84	2.76	4.14
Reduction	*14.4%*	*79.7%*	*69.0%*

Random score perturbation yields 2–5% W_1 change (non-significant), against our 43–89% distributional improvements on the same pools.

F.3 Hyperparameter Sensitivity

λ (relevance–calibration tradeoff). Hybrid methods (W_1-MMR and WassRank OT) each introduce λ. We sweep $\lambda \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ across all three domains (Figure 6). W_1-MMR holds steady at $\lambda \leq 0.5$ on Seller Forums and Yelp, then degrades once relevance dominates. OpinRank behaves differently: λ=0.7 is optimal there (denser pools tolerate more calibration weight). WassRank OT is nearly λ-invariant on dense pools: Yelp gives $W_1 \approx 2.52$ for $\lambda \leq 0.5$ (identical to 3 decimal places) and OpinRank yields W_1=0.95 with 100% EM across that range.[2] No per-domain tuning needed.

[2]The λ-sensitivity results use entity-specific FAISS indices (EM=100%), whereas Table 1 uses a single shared FAISS index across all OpinRank entities (EM=82.9%). This accounts for the OpinRank WassRank OT W_1 difference (3.87 in Table 1 vs. 0.95 here).

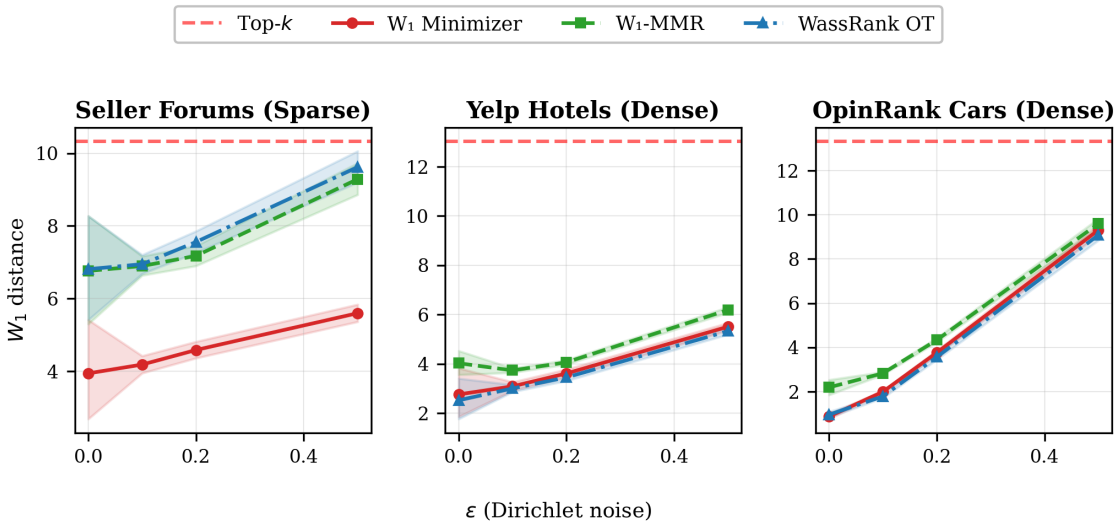

Figure 4: Dirichlet perturbation robustness across all three domains. Dashed red line = Top-k baseline. At ε=0.2, W_1 Minimizer remains 72% better than Top-k on Yelp.

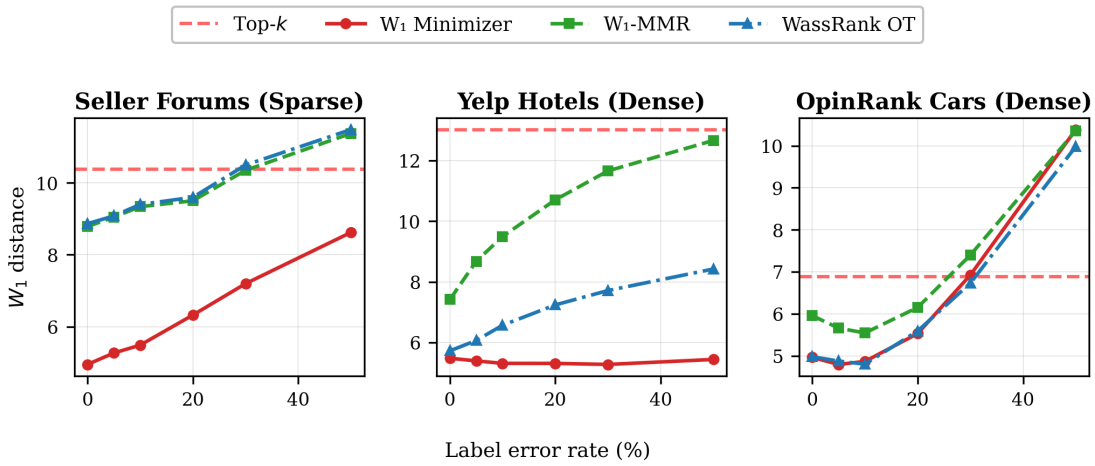

Figure 5: Mislabel sensitivity: W_1 vs. label error rate. Dashed red line = Top-k baseline. W_1 Minimizer never crosses Top-k even at 50% error on Yelp.

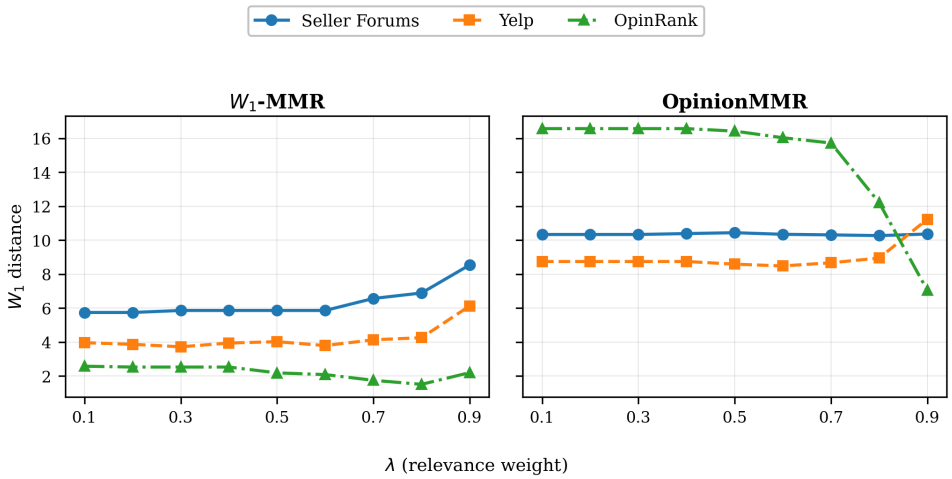

Figure 6: λ-sensitivity for W_1-MMR (left) and OpinionMMR (right). W_1-MMR is stable at $\lambda \leq 0.5$ and degrades at λ=0.9. OpinionMMR shows higher W_1 across all domains with limited sensitivity to λ.

Pool size N. We vary $N \in \{25, 50, 100, 200\}$ with fixed k=20 (Figure 7). N=100 suffices for near-optimal performance on dense pools. W_1-MMR benefits more from larger pools because it draws from the full (unfiltered) candidate set. On OpinRank, even N=50 approaches the N=200 re-

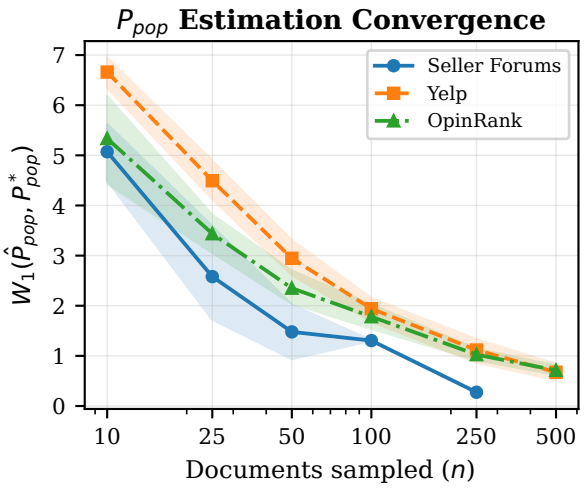

Figure 8: $W_1(\hat{P}_{\text{pop}}, P^*_{\text{pop}})$ vs. entity-matched documents sampled. With 50–100 documents per entity, estimation error is comparable to or below the re-ranking improvement magnitude itself.

sult because entity-specific FAISS indices guarantee most candidates are already entity-matched. On sparse Seller Forums, larger pools become essential for W_1-MMR to locate cross-entity documents that fill distributional gaps.

F.4 Population Estimation and Cold-Start Robustness

We evaluate whether P_{pop} can be reliably estimated from indexed data and whether the system degrades safely at cold-start. For each entity e, the full-corpus distribution $P^*_{\text{pop},e}$ serves as our oracle evaluation target. We sample subsets of $n \in \{10, 25, 50, 100, 250, 500\}$ entity-matched documents, compute $\hat{P}^{(n)}_{\text{pop},e}$, and measure $W_1(\hat{P}^{(n)}_{\text{pop},e}, P^*_{\text{pop},e})$. Crucially, the re-ranker uses $\hat{P}_{\text{pop}}$ but selected evidence is evaluated against P^*_{pop}; this avoids circularity.

Estimation convergence. Figure 8 shows estimation error vs. sample size across all three domains. Convergence is rapid. At n=50, mean W_1 between estimated and oracle distributions drops below 3.0; at n=100, below 2.0. Bootstrap CIs (95%) narrow monotonically, which indicates stable estimation with modest data.

Cold-start behavior. When an entity has few labeled documents, the entity-level P_{pop} estimate gets noisy. We simulate this by capping observation count at $n \in \{10, 25, 50, 100\}$. Even at n=10, all estimation strategies beat uncalibrated Top-k. Smoothed interpolation with a domain prior ($\tilde{P} = \alpha\hat{P}_e + (1-\alpha)P_{\text{domain}}$, $\alpha = n/(n+50)$) works best at every count, achieving 61% W_1 reduction at n=10. Our recommended production rule: use entity-level P_{pop} when $n \geq 25$ and at least 2 sentiment bins have mass > 0.05; otherwise fall back to the smoothed domain prior.

G Generation Evaluation

G.1 Judge Validation & Positional Bias

Generation evaluation should not hinge on a single model's quirks; leniency bias and prompt sensitivity are well-documented failure modes (Thakur et al., 2025). We validate with five independent model families from different providers: Claude Sonnet 4 (Anthropic), Llama 3.3 70B (Meta), Mistral Large 3 (Mistral AI), Amazon Nova Pro (Amazon), and DeepSeek v3.2 (DeepSeek). Appendix A.3.3 contains the full judge prompt with its three evaluation dimensions. Below we confirm positional bias, report directional inter-judge agreement, and present majority-vote consensus.

G.1.1 Positional Bias

LLM judges exhibit positional bias, a preference for whichever answer appears first (Wang et al., 2024; Shi et al., 2025). We confirm this across all five judges in our panel. Our position-swap protocol converts inconsistent position-dependent preferences into conservative ties, motivating the directional κ_d analysis in §G.1.2. Bias rates differ across models, producing asymmetric tie distributions:

Table 18: Consensus tie rates across the 5-judge panel. Higher tie rates indicate stronger positional bias (position-swap converts inconsistent preferences to ties).

Judge	Consensus Tie Rate (k=5)	Consensus Tie Rate (k=10)
Claude Sonnet 4	46%	46%
Mistral Large 3	48%	61%
DeepSeek v3.2	63%	71%
Llama 3.3 70B	73%	86%
Amazon Nova Pro	79%	86%

When judges do commit to a preference, they agree directionally 85–100% of the time.

G.1.2 Directional Inter-Judge Agreement (κ_d)

We want to separate genuine directional disagreement from artifacts of unequal tie rates. So we compute κ_d over only those pairs where both judges commit to a non-tie preference:

All five judges agree directionally at κ_d=0.61–1.00. Among the four non-Anthropic judges, pairwise κ_d ranges 0.95–1.00 at both k values.

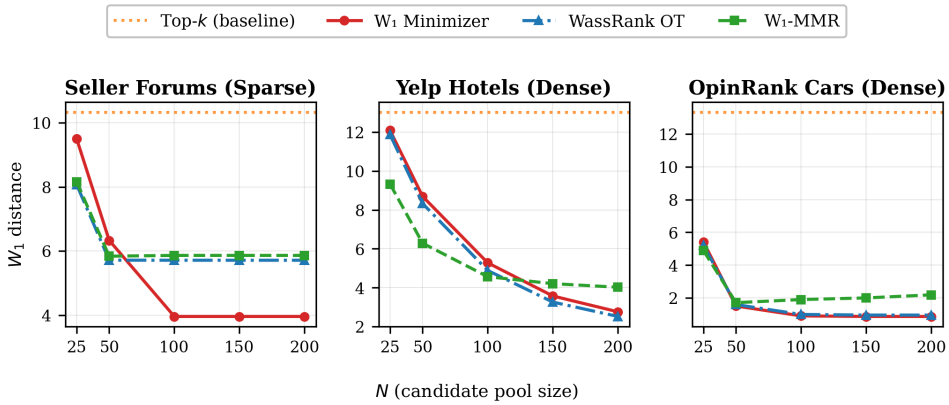

Figure 7: W_1 vs. pool size N (k=20). N=100 suffices for dense pools; W_1-MMR benefits from larger pools on sparse data.

Table 19: Mean directional κ_d per judge vs. all others (fairness dimension). Non-Anthropic judges agree at $\kappa_d \geq 0.95$ at k=10; at k=5, κ_d=0.88–0.93.

Judge (vs. others)	k=5 κ_d	k=5 **Agree%**	k=10 κ_d	k=10 **Agree%**
Claude vs. others	0.68	88.2%	0.86	93.5%
Llama 3.3 vs. others	0.89	96.0%	0.97	98.8%
Mistral L3 vs. others	0.89	95.8%	0.95	97.8%
DeepSeek vs. others	0.93	97.4%	0.95	97.9%
Nova Pro vs. others	0.88	95.3%	0.94	97.3%

Claude's lower κ_d (0.68/k=5, 0.86/k=10) traces to its lower tie rate: it commits more often and occasionally flags nuances others collapse into ties.

G.1.3 Majority Vote Results (3/5 Judges Must Agree)

Under majority vote (3/5 agreement required), calibration methods reach strong significance at k=5 while Random correctly fails. This is a critical sanity check. At k=10, fewer approaches survive; context-window averaging attenuates distributional differences. We extend to k=20 under the same protocol; the scaling regime distinguishes semantic-diversity from distributional methods.

At k=5 with only 2–3 active sentiment bins, diversity and calibration objectives partly overlap and MMR/OpinionMMR win 83%/82% (Table 20). By k=20 the pool has room for finer-grained proportions and pure semantic-diversity has run out of signal; MMR and OpinionMMR Fair% drop to 29% and 33%. Distributional methods maintain their advantage: WassRank OT rises from 70% (k=5) to 73% (k=10) to 92% (k=20) as budget lets proportional targeting express itself in the answer. DPP is a partial exception: it also grows with k (89% at k=20) because its determinantal spread lands one representative per bin when there is enough budget.

Domain decomposition at k=10 reveals a clear density dependency. On dense Yelp (60 questions, 10 aspects), all calibration methods achieve 86–97% fairness decided rate; W_1 Minimizer hits 97% (28W/1L). Dense OpinRank (60 questions, 10 entities) confirms propagation: DPP leads at 100% (19W/0L), W_1 Minimizer at 94% (16W/1L), and 5 of 8 approaches reach $p < 0.001$. Sparse Seller Forums (36 questions, 6 entities, EM∼40%) is the outlier: only OpinionMMR achieves significance (10W/2L, 83%, p=0.019); W_1 Minimizer and W_1-MMR hover near coin-flip (54–55%), confirming that sparse pools cap document quality regardless of re-ranking strategy. Random fails on all three domains (13–53%), validating the sanity check.

G.1.4 Judge Prompt Sensitivity

Our main fairness criterion conflates two signals: *proportional accuracy* and *viewpoint coverage*. We disentangle them by re-judging all answer pairs under two contrasting definitions (Prompt A and Prompt B in Appendix A.3.3), same 5-judge panel with position-swap (Zheng et al., 2023) and 3/5 majority vote:

- **Prompt A (Proportional Accuracy):** Judges receive the ground-truth P_{pop} and are asked which answer's emphasis better matches the actual distribution. An answer over-representing a 10% minority relative to a 70% majority is penalized as misleading.
- **Prompt B (Viewpoint Coverage):** Judges are asked which answer covers more distinct viewpoints, including rare ones. Giving disproportionate space to minority views is rewarded.

INFORMATIVENESS and GROUNDEDNESS dimensions remain identical across both prompts, serving as controls.

Three patterns emerge. (1) Under proportional accuracy, W_1 Minimizer leads DPP by 8/6 pp at

Table 20: 5-judge majority-vote generation results (cross-domain, 156 questions) at $k \in \{5, 10, 20\}$. All calibration methods achieve $p < 0.001$ at k=5; retain significance at k=10 ($p \leq 0.021$) and k=20 ($p \leq 0.004$). Random fails at all k values. At k=20, MMR/OpinionMMR Fair% collapse (29%/33%) as semantic diversity saturates; distributional methods maintain or grow their advantage; WassRank OT rises 70%→73%→92%. Fair% = W/(W+L). Win% = W/N.

	k=5			k=10			k=20		
Approach	**W/L/T**	**Fair%**	**Win%**	**W/L/T**	**Fair%**	**Win%**	**W/L/T**	**Fair%**	**Win%**
Random	16/36/104	31%	10%	12/40/104	23%	8%	3/26/127	10%	2%
MMR**	39/8/109	83%	25%	21/9/126	70%	13%	5/12/139	29%	3%
OpinionMMR**	50/11/95	82%	32%	25/8/123	76%	16%	7/14/135	33%	4%
DPP**	54/7/95	89%	35%	42/6/108	88%	27%	25/3/128	89%	16%
WassRank OT**	66/28/62	70%	42%	75/28/53	73%	48%	33/3/120	**92%**	21%
W_1-MMR**	44/10/102	81%	28%	24/7/125	77%	15%	13/2/141	87%	8%
W_1 Minimizer**	51/8/97	86%	33%	32/8/116	80%	21%	27/5/124	84%	17%

Table 21: Judge prompt ablation: fairness win rate (%) under proportional vs. coverage definitions (3 domains, 156 questions per approach). W_1 Minimizer dominates under proportional accuracy; the gap narrows under coverage. Stratified sampling (oracle) confirms the Prompt A criterion tracks the ground-truth distribution, not any specific retrieval strategy — W_1 Minimizer matches oracle Fair% within 1 pp at k=5.

	k=5		k=10	
Approach	**Prop. (A)**	**Cov. (B)**	**Prop. (A)**	**Cov. (B)**
Stratified (oracle)	35%	—	—	—
W_1 Minimizer	34%	38%	30%	27%
DPP	26%	35%	24%	25%
W_1-MMR	23%	36%	18%	17%
OpinionMMR	21%	30%	13%	19%
MMR	13%	26%	10%	13%
Random	16%	13%	12%	7%

Table 22: Control dimensions remain stable across prompt variants (k=5, Δ = Prompt B − Prompt A). Shifts ≤4 pp confirm the fairness prompt change does not contaminate other evaluation axes.

Approach	**Info% (A)**	**Info% (B)**	**Grnd% (A)**	**Grnd% (B)**
W_1 Minimizer	44%	40%	28%	24%
DPP	38%	40%	27%	24%
Random	19%	18%	15%	8%

k=5/10 and sits within 1 pp of the stratified oracle (35%), confirming Prompt A tracks ground-truth P_{pop} rather than any retrieval strategy. (2) Under viewpoint coverage, the gap narrows to 3/2 pp; DPP's diversity edge is real but modest once W_1's calibration already covers most bins. (3) Inter-judge agreement rises under Prompt B (Llama–Mistral κ: 0.39→0.62 at k=5), suggesting viewpoint counting is more objectively evaluable without ground truth. Our original conflated criterion thus understates W_1's advantage at its design objective.

G.2 Single-Judge Metric Isolation (Generation)

The 5-judge majority-vote protocol is conservative — positional-bias-driven ties inflate the tie column. To isolate the ordinal metric's end-to-end contribution with more statistical power, we ran a controlled single-judge (Claude Sonnet 4) position-controlled pairwise evaluation between W_1 Minimizer and KL(JS) Minimizer at k=5 (Table 23). Same algorithm, same entity-filtered pool; only the distance function differs.

Table 23: W_1 Minimizer vs. KL(JS) Minimizer, generation evaluation at k=5 (156 queries, single-judge Claude Sonnet 4, position-controlled, blind pairwise vs. Top-k). W_1 leads KL(JS) by +7 pp cross-domain; both significantly beat Top-k (p<0.001).

Domain	**Method**	N	**W/L/T**	**Fair%**	**Win%**
Seller Forums	W_1 Minimizer	36	10/9/17	**53%**	**28%**
	KL (JS) Minimizer	36	7/10/19	41%	19%
Yelp	W_1 Minimizer	60	39/4/17	**91%**	**65%**
	KL (JS) Minimizer	60	31/8/21	79%	52%
OpinRank	W_1 Minimizer	60	29/8/23	78%	48%
	KL (JS) Minimizer	60	35/10/15	78%	**58%**
Cross-domain	W_1 Minimizer	156	78/21/57	**79%**	**50%**
	KL (JS) Minimizer	156	73/28/55	72%	47%

W_1 leads KL(JS) by +7 pp Fair% cross-domain; the retrieval-side advantage (Appendix C.1) does not invert at generation. Yelp shows the largest gap (91% vs. 79%); Seller Forums the same direction at smaller magnitude (53% vs. 41%). OpinRank is a partial exception (Fair% ties; KL(JS) edges Win Rate 58% vs. 48%) — its near-uniform P_{pop} compresses ordinal distances, leaving the ordinal ground cost less to do.