Does the Label Space Need to Evolve? Bounding the Gains of Self-Improving Context Engineering for In-Context Classification

Anonymous authors
Paper under double-blind review

Abstract

Can a classifier improve by rewriting its label space during inference? We study this question with EvoLabel, which separates a prediction architecture from an optional evolution layer. The predictor compares two retrieval views over a fixed labeled pool, returns immediately when their canonical predictions agree, and invokes a configured arbiter otherwise. A coverage gate avoids the second view when leave-one-out retrieval is unreliable. The evolution layer aggregates prediction-pair feedback and may propose structured Merge/Describe updates to the presented label state, without changing model weights or adding labeled examples. In a 24-method, 12-dataset, three-seed comparison, the adaptive retrieval–arbitration configuration reaches 81.1% mean accuracy, versus 79.2% for the strongest external baseline; on four many-class intent datasets the corresponding result is 86.0% versus 83.5%. The same configuration exceeds the frozen dual-arm baseline on three additional models by 1.38–1.75 percentage points. On the intent tasks, agreement covers 86.3% of queries and identifies a substantially more accurate subset (91.3% versus 52.6% on disagreements), although this contrast is descriptive rather than a calibrated risk guarantee. Crucially, the evolution-enabled runs in the replication commit no label-space edits: the observed gains support the prediction architecture, not successful online evolution. A conditional decomposition expresses rewrite gain as affected-query mass times conditional accuracy change under an explicit outside-subset invariance assumption. Together with a frequency-matched operator audit, this explains what the null does and does not establish. The study provides an audited, trace-backed comparison and a bounded claim: complementary views and arbitration help on this suite, while unseen-dataset validation and more effective rewrite operators remain open.

1 Introduction

A growing body of work treats an LLM's context not as a fixed prompt but as a *non-parametric artifact* that a system may rewrite from its own feedback: ACE (Zhang et al., 2026) maintains an evolving free-text *playbook* updated by generation–reflection–curation and reports double-digit gains on multi-step agent benchmarks, and the natural extrapolation is that *any* context artifact should benefit from online evolution. This paper asks whether that extrapolation holds for the *label space* of an in-context classifier—the set of candidate labels enumerated in the prompt. Label presentation is a first-class lever (Chen et al., 2023; Wang et al., 2023a; Zheng et al., 2024), yet it is treated statically; online adaptation is pursued on the *selection* axis (which exemplars to retrieve) (Xu et al., 2026). To our knowledge, prior work has not evaluated whether this presentation axis should evolve online. We report a null for the specific mechanism tested and do not infer that label-space evolution fails in general (Section 5).

The question is not vacuous. A label space is the one context artifact whose edits are cheap, structured, and directly legible to the model: merging two near-synonymous classes or sharpening a description changes what the classifier is asked to choose among, without touching the exemplars or the weights. These operations restrict the form of a proposed edit and make its immediate effect measurable, while still requiring validation against harmful semantic drift.

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.

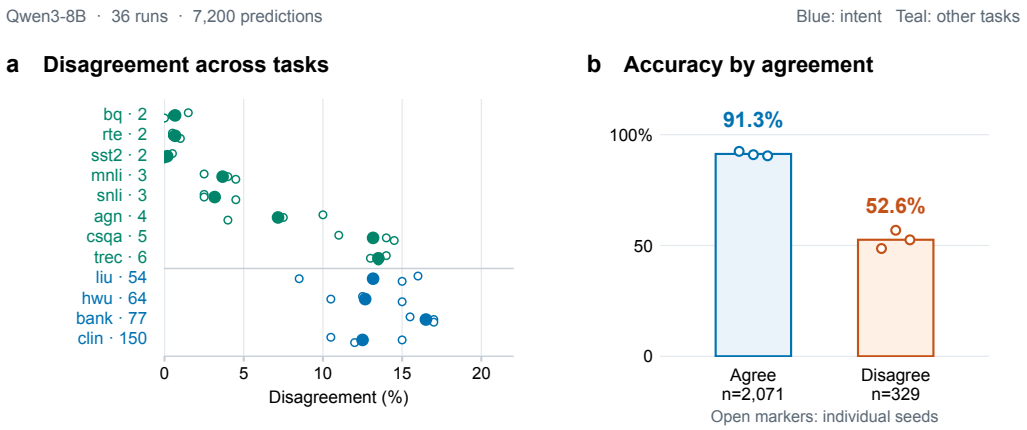

Figure 1: **Disagreement isolates a difficult decision slice.** (a) Pooled disagreement on twelve tasks; open dots show individual seeds and blue denotes intent tasks. (b) On intent tasks, final accuracy is 91.3% on agreement and 52.6% on disagreement. Task family covaries with cardinality, so panel (a) is descriptive.

Why the arbitration step matters. Two selectors run on the same input disagree on a non-trivial slice: in the Qwen3-8B six-ballot runs, the pooled disagreement rate is 5.3% on eight datasets with 2–6 classes and 13.7% on four intent datasets with 54–150 classes. On the latter, final accuracy is 52.6% on disagreed queries versus 91.3% on agreed queries (Figure 1). This is a descriptive contrast: task family and label cardinality vary together. Every disagreed item is decided by the *arbiter*, so the arbitration rule is a first-class design decision—and the object we study.

We evaluate a dual-arm predictor together with a label-space reflector. The evidence separates three questions: whether complementary prediction views help, whether their disagreement identifies difficult queries, and whether label rewriting adds value beyond that architecture. The first two receive empirical support; the tested evolution mechanism commits no edits in the replication. Our contributions are:

1. An audited, trace-backed comparison of retrieval and arbitration configurations on 12 datasets and four model backbones, with an explicit benchmark-calibrated adaptive rule.
2. A trace-based analysis of disagreement, selective prediction, and held-out confusion-pair transfer, separating prediction gains from rewrite effects.
3. A conditional gain decomposition and an empirical null for the tested label-evolution mechanism, with its assumptions and scope stated explicitly.

2 Related Work

In-context learning and example selection. Language models can use demonstrations without updating their parameters (Brown et al., 2020), a paradigm whose mechanisms have been extensively surveyed (Dong et al., 2024). Their effectiveness depends on what the examples communicate, not simply the number of labels shown (Min et al., 2022). Similarity-based retrieval (Liu et al., 2022), diversity-aware demonstrations (Levy et al., 2023), selective annotation (Su et al., 2023), and instance-specific selection and ordering (Wu et al., 2023) address complementary aspects of this problem. Selection strategies also vary with the data and model (Peng et al., 2024), motivating efficient alternatives such as gradient-based selection (Zhang et al., 2025b), reinforcement-learning-based selection (Wang et al., 2025), and meta-learning-based selection (Wang & Jia, 2026). We keep the evidence pool fixed and compare concrete retrieval configurations; the experiments do not establish that presentation changes generally dominate selection methods.

Label presentation, calibration and model selection. Knowledge-informed verbalizers (Hu et al., 2022), the information-flow role of label tokens (Wang et al., 2023a), and label-space manipulation

(Chen et al., 2023) motivate treating the label inventory as an explicit context artifact. Candidate ordering and prompt permutations can change outcomes (Zheng et al., 2024; Lu et al., 2022); contextual calibration addresses a related answer-bias problem (Zhao et al., 2021). These findings motivate controlled presentation, but do not imply that repeated rewriting improves accuracy. Moreover, few-shot evaluation must account for the data used to select prompts and hyperparameters (Perez et al., 2021), and collaborative rationale-driven prompting has been explored for annotation (Wu et al., 2025). Our routing threshold is calibrated on the reported suite, a limitation that remains even though inference does not read test labels.

Evolving context and expert routing. ACE maintains and updates contextual knowledge (Zhang et al., 2026), while Dynamic Cheatsheet retains reusable information across queries (Suzgun et al., 2026); both fit the broader context-engineering perspective (Mei et al., 2025). Prior work also evolves prompt text directly via LLM-driven mutation (Fernando et al., 2024) or discrete evolutionary search (Guo et al., 2024). Our label inventory is a constrained artifact, and the reported null concerns its tested update mechanism rather than these broader approaches. SCOTTA studies confidence-guided online adaptation in multi-label classification (Xu et al., 2026); our SCOTTA-style comparator is an adaptation to the present single-label protocol, not a reproduction of every original setting. RouteLLM learns a router from preference data (Ong et al., 2025); our cardinality rule is a fixed system configuration, not a learned query router.

Ensembles and selective prediction. Multi-view disagreement (Blum & Mitchell, 1998) and sampled-answer aggregation (Wang et al., 2023b) motivate alternative views, but do not guarantee independent errors or useful corrections. Rejection has a long history (Chow, 1957), including post-hoc selective classification (Geifman & El-Yaniv, 2017), integrated reject options (Geifman & El-Yaniv, 2019), and trust estimates (Jiang et al., 2018). Language-model calibration (Jiang et al., 2021), semantic entropy (Kuhn et al., 2023), and Bayesian prompt ensembles (Tonolini et al., 2024) provide additional uncertainty approaches. We evaluate a simpler agreement indicator already available from two predictions. It is neither a calibrated probability nor a claimed replacement for trained selective models. Appendix A.13 expands these distinctions and their implications for the evaluation.

3 From Static to Evolving Label Spaces

Let $P = \{(x_j, y_j)\}_{j=1}^{n}$ be a fixed labeled retrieval pool and $\mathcal{Y}$ the canonical evaluation inventory. We represent the presented label state as

$$L_t = (\mathcal{Z}_t, \pi_t, d_t), \qquad \pi_t : \mathcal{Z}_t \to 2^{\mathcal{Y}} \setminus \{\varnothing\}, \quad \bigcup_{z \in \mathcal{Z}_t} \pi_t(z) = \mathcal{Y}, \tag{1}$$

where $\mathcal{Z}_t$ is the set of displayed entries, π_t records the canonical classes represented by an entry, and d_t supplies optional descriptive text. Initially π_0 is one-to-one. $\textsc{Merge}(a, b)$ replaces two presentation entries by their union; $\textsc{Describe}$ changes d_t using prototypical surface forms from P. A canonical decoder ρ_t resolves model outputs to $\mathcal{Y}$ before agreement and scoring. Thus a presentation merge does not merge evaluation classes. Neither operator changes P, obtains a new annotation, or updates model parameters.

This distinction makes the scientific question precise. Label-state evolution can alter the prompt and the demonstrations retrieved for later queries, even though the underlying evidence and evaluation task remain fixed. Consequently, the absence of new labels does not by itself imply invariant predictions from a finite model. Unlike an evolving free-text playbook (Zhang et al., 2026), the state and its legal changes are structured and auditable. We evaluate the dual-view prediction architecture and the optional rewrite layer separately (Figure 2), so an architectural gain is not attributed to an operator that never executes.

4 Method: EvoLabel

4.1 Fixed-pool prediction and the coverage gate

Let $\mathcal{R}_v(x; P, L, k)$ return k demonstrations for retrieval view $v \in \{F, R, M\}$. Arm F uses embedding top-k under L_0; R promotes semantic diversity under L_t; M uses BM25 lexical matching. Sentence

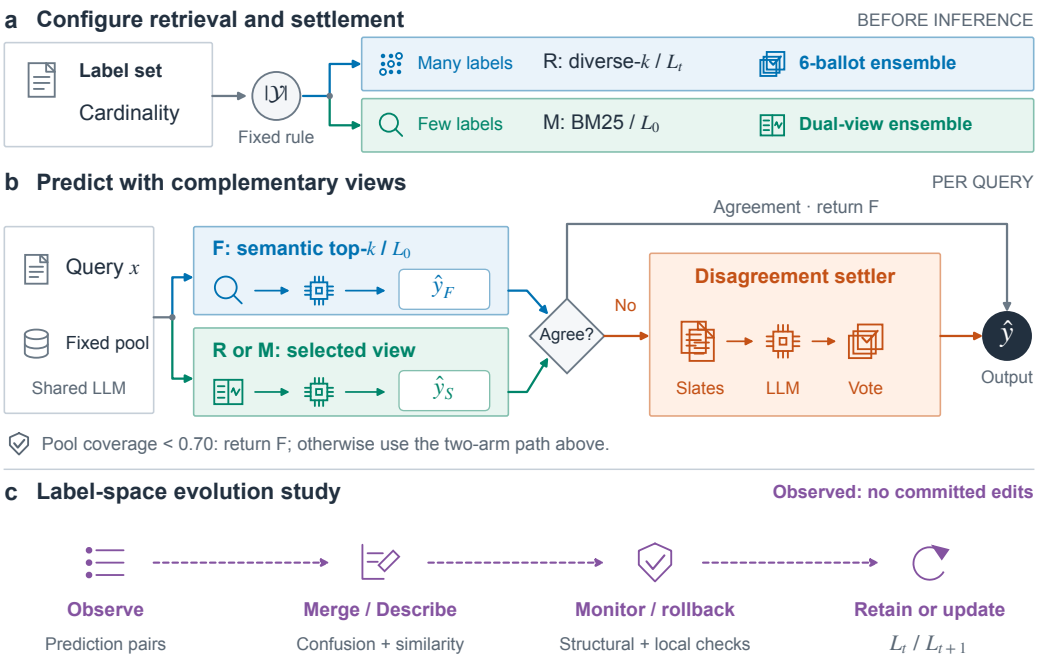

Figure 2: **Architecture overview.** (a) A fixed cardinality rule selects the second retrieval view and arbiter ($|\mathcal{Y}| \geq \tau$: R; otherwise: M); the cut τ is benchmark-calibrated and recorded in Appendix A.11. (b) With the coverage gate open, agreement returns F; disagreement triggers arbitration. A closed gate uses F alone. (c) Dashed arrows denote the separately evaluated evolution layer. Its enabled runs in the replication committed no edits.

embeddings are provided by Sentence-BERT (Reimers & Gurevych, 2019). For a fixed LLM h_θ, the canonical prediction of a view is

$$p_v(x, L) = \rho_L[h_\theta(x, \mathcal{R}_v(x; P, L, k), L)] \,. \tag{2}$$

All views share P, h_θ, decoding, and output classes. They vary only the construction of in-context evidence and its presentation.

The second arm is enabled only when the fixed pool supports reliable local retrieval. Define the leave-one-out any-hit rate

$$C(P) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{1}\left[y_j \in \{y : (x, y) \in \mathcal{R}_F(x_j; P \setminus \{(x_j, y_j)\}, L_0, k)\}\right], \quad g = \mathbf{1}[C(P) \geq 0.70]. \tag{3}$$

The gate is computed before test inference from pool labels only, without test labels or LLM calls; it decides whether a second view is queried and is not a certificate that later rewrites preserve predictions.

4.2 Dual-view agreement and arbitration

When the gate is open, a dataset-level router pairs a *challenger* view S with an *arbiter* A before any test query is processed:

$$(S, A) = \begin{cases} (R, A_6), & |\mathcal{Y}| \geq \tau, \\ (M, A_{12}), & |\mathcal{Y}| < \tau, \end{cases} \qquad \hat{y}(x) = \begin{cases} p_F(x, L_0), & g = 0 \text{ or } p_F = p_S, \\ A(x, P, p_F, p_S), & g = 1 \text{ and } p_F \neq p_S. \end{cases} \tag{4}$$

The predicate separates two expert configurations rather than fitting a threshold: the two families' label-space cardinalities do not overlap, so any cut in the intervening interval induces the same assignment. The cut τ is fixed before test inference and calibrated on this suite (value and invariance interval: Appendix A.11); these data cannot validate it.

Agreement is evaluated on canonically decoded labels rather than raw generations: equality is tested only after ρ_L maps both outputs to $\mathcal{Y}$ (Eq. 2), so surface-form variation does not inflate the disagreement rate. Equation 4 thus defines both the prediction rule and the disagreement indicator $D(x) = \mathbf{1}[p_F \neq p_S]$ used in Section 7.

The arbiter aggregates *re-retrieved evidence* rather than resampled generations: A_b constructs b independent demonstration slates, records one canonical vote per slate, and returns the majority label, breaking ties with p_F. A_6 varies query relevance across six candidate-focused slates; A_{12} combines eight semantic and four lexical slates so that neither retrieval family dominates (Appendix A.5).

Prediction and evolution have separate state boundaries: the scored output for x_t is finalized before feedback reaches the reflector (Appendix A.14).

4.3 The reflector

For window W_t, the reflector records directed pair support

$$N_t(a, b) = \sum_{i \in W_t} \mathbf{1}[p_F(x_i, L_0) = a,\ p_S(x_i, L_i) = b], \qquad |W_t| \leq 64. \tag{5}$$

Support and label-name similarity rank candidate Merge/Describe operations. A proposal is eligible only after support four; a run permits at most eight operations. If $q(L_t, o_t) \in \{0, 1\}$ denotes the implementation's type and state-consistency checks, the state transition is

$$L_{t+1} = \begin{cases} \Phi(L_t, o_t), & q(L_t, o_t) = 1, \\ L_t, & \text{otherwise.} \end{cases} \tag{6}$$

The trace records o_t, q, and the resulting state. These checks prevent malformed or repeated operations; they do not guarantee accuracy preservation outside the triggering pair.

The label-free variant supplies prediction-pair feedback only. Supervised diagnostic variants additionally reveal the gold labels of 10% or 100% of the queries in a round, and so test a different question—whether direct error evidence can activate the operators—and are excluded from deployment claims. In the evolution-enabled replication records no candidate is committed, hence $L_t = L_0$ throughout (variants and operation counts: Appendix A.8).

4.4 Variants and the adaptive router

The adaptive configuration fixes the challenger–arbiter pair of Eq. 4 before any test query is processed. Selected on this suite, it is a source of optimism, and because the route changes challenger and arbiter jointly, a comparison with a fixed expert is an end-to-end configuration comparison rather than a single-module ablation. Cross-model evaluation tests backbone transfer, not unseen-dataset validation; the variant catalog is in Appendix A.15.

4.5 Cost accounting

Arbiter cost is proportional to the disagreement rate: with $D(x)$ from Eq. 4 and $b_A \in \{6, 12\}$, each query costs $K(x) = 1 + g\{1 + D(x)b_A\}$ prediction-path calls—one with a closed gate, two on agreement, and $2 + b_A$ on disagreement. Label-free execution needs no test labels or log-probabilities, and logged runtime is descriptive rather than hardware-normalized (Appendix A.5).

5 A Conditional Gain Decomposition

Fix a baseline system s_0, a modified system s_1, and a query subset D with mass δ. Taking D to be the baseline disagreement set is useful only if the following condition holds.

Assumption 5.1 (Outside-subset invariance)**.** *For every query outside D, $s_1(x) = s_0(x)$.*

Agreement bypasses arbitration on the current query, but this does not guarantee invariance after a future rewrite. We use the assumption conditionally; it is not implied by a coverage threshold or by calling an operator a re-encoding.

Proposition 5.2 (Evolution Gain Decomposition)**.** *Under Assumption 5.1, let* $\Delta_D = \Pr[s_1 = y \mid D] - \Pr[s_0 = y \mid D]$. *Then*

$$\mathrm{acc}(s_1) - \mathrm{acc}(s_0) = \delta\Delta_D \le \delta\{1 - \Pr[s_0 = y \mid D]\}.$$

Proof. Condition on D and D^c. The contribution on D^c vanishes by invariance; conditional accuracy on D is at most one. □

This separates reach from efficacy without asserting an empirical efficacy is a universal upper bound. For the earlier seven-variant diagnostic pool, $\delta = 13.44\%$ and the reported conditional contrast is 2.22 pp, giving 0.30 pp descriptively; an oracle correct on every affected query instead permits roughly 7.0 pp. Neither calculation proves that all rewrite mechanisms must fail. Multi-round extensions and limitations appear in Appendices A.1–A.2.

6 Benchmark

Datasets. We evaluate twelve text-classification tasks in two families: four intent benchmarks with large, semantically overlapping inventories—banking77 (Casanueva et al., 2020) (76 labels in the retrieval pool), clinc150 (Larson et al., 2019) (144–147), HWU64 (64) and Liu54 (50) (Liu et al., 2021)—and eight tasks with 2–6 classes—AG-News, SST-2, TREC, RTE, SNLI and MNLI (Bowman et al., 2015; Williams et al., 2018), BoolQ (Clark et al., 2019), and CommonsenseQA (Talmor et al., 2019). Intent classification is the canonical presentation-side setting: its large inventories make retrieval *against* the label space a material part of the decision. Per-task label counts, split sizes, sources and licences are given in Appendix Table 8, and the landscape is summarised in Figure 6a.

Baselines. The main grid contains two dual-arm systems and 22 comparators spanning five families: direct retrieval, diversity and prototype selection, online and uncertainty-based selection, and standardized catalog adapters (Appendix Table 9, Figure 6b). All methods share the pool, prompt, k and decoding, differing only in demonstration selection or the named dual-arm machinery. Catalog identifiers denote controlled local implementations rather than claims of exact reproduction.

Protocols. The main grid uses Qwen3-8B (Yang et al., 2025) with $k = 16$, a fixed pool of 500 training examples, 200 test queries and seeds 42–44: 24 methods yield 864 runs. Three additional backbones—Llama3.1 (Dubey et al., 2024), Mistral (Jiang et al., 2023) and Qwen2.5 (Qwen et al., 2024)—use the same twelve tasks and three seeds. Training pools and test queries are disjoint. Earlier evolution diagnostics and the extended arbiter study use different protocols and are reported separately. Hyperparameters, prompt templates, decoding settings and the statistical unit are specified in Appendices A.3–A.6. We report accuracy, seed variability, disagreement and operation counts; repeated queries across seeds are not independent observations, so pooled query-level significance is interpreted with that limitation.

Query-sample caveat. Each method is scored on its own 200-query sample per seed. On five tasks (AG-News, SST-2, TREC, SNLI, MNLI) the archived runs contain two distinct query samples per seed, so per-task figures on those tasks are not query-paired across *all* methods; paired tests are therefore restricted to method pairs sharing a sample. For the headline Adaptive-versus-SCOTTA comparison ten of the twelve tasks share a sample, whereas SNLI and TREC do not and are excluded from the paired analyses (Appendix A.4).

7 Experiments

7.1 Main comparison and task-level variation

Table 1 reports the complete 24-method grid, including the train-order dual-arm comparator. EvoLabel-Adaptive reaches 81.1% overall versus 79.2% for SCOTTA-style, the highest-average external baseline; on the four intent datasets the means are 86.0% and 83.5%. Taking the dataset as the unit of analysis and restricting to the ten tasks on which the two methods share a query sample (Section 6), the +1.6 pp gap has a dataset-clustered bootstrap 95% interval $[+0.17, +2.75]$ pp and Wilcoxon $p = 0.047$, with Adaptive ahead on nine of ten; over all twelve tasks the descriptive gap is +1.9 pp with Adaptive ahead on eleven of twelve, but SNLI and TREC are not query-paired there

Table 1: **Main comparison.** Accuracy (%) across 12 datasets, averaged over three seeds (Qwen3-8B; 500 training examples, 200 queries, $k = 16$). All 24 methods and 864 runs are included. Bold marks column maxima; seed variability is reported in Table 10.

Method	bank	clin	hwu	liu	agn	sst2	trec	rte	snli	mnli	bq	csqa	Mean
EvoLabel-Adaptive	**84.5**	**89.0**	**87.3**	**83.2**	**86.7**	**94.8**	**77.3**	85.2	**82.2**	**78.2**	64.2	61.2	**81.1**
Train-order dual-arm	83.3	88.2	86.7	82.0	86.5	94.3	73.2	82.8	79.8	74.7	**64.5**	60.8	79.7
scotta_style	81.3	84.8	86.3	81.3	84.7	94.2	72.3	83.3	79.8	74.3	63.3	**64.7**	79.2
static_topk	80.8	85.5	84.0	81.0	84.8	94.2	71.2	82.7	78.3	73.0	64.2	64.0	78.6
uncertainty	76.7	83.3	82.0	80.2	83.0	94.2	74.8	82.3	79.8	71.3	60.5	61.2	77.4
tfidf_knn	76.2	82.5	81.8	79.7	82.8	93.8	75.8	81.5	79.3	71.2	59.8	60.8	77.1
dpp_diversity	75.3	83.5	81.0	79.7	83.2	93.8	75.7	82.0	78.3	71.5	60.0	60.0	77.0
bm25	71.2	84.0	80.2	77.3	83.8	94.0	76.0	82.8	76.3	73.3	60.5	61.3	76.7
diverse_cone	72.0	79.3	77.5	75.0	83.0	92.7	71.7	**85.5**	74.8	72.7	62.0	61.8	75.7
kmeans_prototype	72.7	80.5	78.5	78.3	83.2	93.5	63.3	82.5	77.2	74.0	62.0	62.2	75.7
topk_cone	72.7	80.0	76.2	77.2	83.8	92.7	62.7	83.2	77.5	73.5	64.3	61.2	75.4
vicl	73.7	79.8	78.8	77.7	83.7	93.5	61.5	82.8	77.2	73.3	60.2	61.0	75.3
label_balanced	71.5	80.8	75.3	66.8	80.5	93.8	76.2	81.8	78.5	71.5	59.7	61.0	74.8
gradmatch_many_shot	73.3	79.0	80.2	77.5	83.5	93.5	57.2	83.5	76.3	71.0	62.5	59.8	74.8
influence_style	71.8	81.2	76.0	66.3	80.5	94.2	76.0	81.0	78.2	71.5	59.3	60.7	74.7
vlm_e2e	73.3	78.2	78.8	76.2	83.3	93.2	54.3	85.0	76.7	73.2	61.3	62.7	74.7
contrastive_icl	72.5	78.7	80.8	76.5	84.0	93.0	50.7	84.2	75.2	72.7	60.5	61.0	74.1
moicl	73.0	78.8	79.0	76.7	83.7	93.0	47.3	83.5	77.2	73.2	61.0	60.0	73.9
hdoo	73.8	77.5	79.2	76.7	82.7	92.7	47.7	84.3	78.3	71.5	60.5	60.7	73.8
mod	71.0	74.3	78.0	73.7	80.7	93.2	57.3	84.0	74.5	71.3	61.3	62.7	73.5
explora	73.8	79.0	78.7	77.0	80.0	92.5	50.7	77.0	76.0	71.8	61.0	59.0	73.0
inficl	73.8	79.0	78.7	77.0	84.3	92.3	46.0	82.0	72.7	69.3	61.8	58.2	72.9
coreset	62.3	70.0	69.0	67.8	79.7	93.5	64.3	83.2	78.2	74.5	58.2	61.5	71.8
vote_k_proxy	72.3	80.8	76.5	68.8	80.2	93.0	43.8	80.8	73.7	69.7	57.3	62.3	71.6

(Appendix A.4). The intent gap is +2.5 pp, interval $[+1.42, +3.67]$ pp. Figure 3a shows paired per-task differences and seed variability: the gain is heterogeneous and CommonsenseQA favours SCOTTA-style. Task family and label cardinality covary, so this does not identify cardinality as the causal source. Panels e–f are descriptive counts rather than independent tests; the complete matrix is in Figure 7.

7.2 Transfer across model backbones

Table 2 retains the cross-model comparison; Figure 3c makes its magnitude visible. Adaptive accuracy exceeds the frozen dual-arm by 1.54, 1.75, and 1.38 pp on Llama3.1-8B, Mistral-7B and Qwen2.5-32B, respectively (differences use unrounded values in Table 15). The margin over the fixed dual-view configuration is much smaller on Mistral and Qwen2.5. Thus, transfer supports the overall configuration but does not establish a uniformly large advantage over every fixed expert. The models share the dataset suite used to calibrate the routing rule; transfer across models is not held-out-dataset validation. Per-dataset results and the reported paired tests remain in the appendix.

7.3 Disagreement and selective prediction

The 36 six-ballot Qwen3-8B runs used in Figure 1 contain 7,200 predictions. On the intent tasks, the arms disagree on $329/2400$ queries; final accuracy is 91.3% on agreement and 52.6% on disagreement. Figure 3d displays the resulting accuracy–coverage tradeoff for every task rather than only a pooled improvement. Agreement uses no additional calls beyond the two views, but the dual-arm system itself is not cost-free relative to a single-arm baseline. Results for the earlier two-seed variant pool are reported separately in Table 21; their protocol and values must not be conflated with these six-ballot runs.

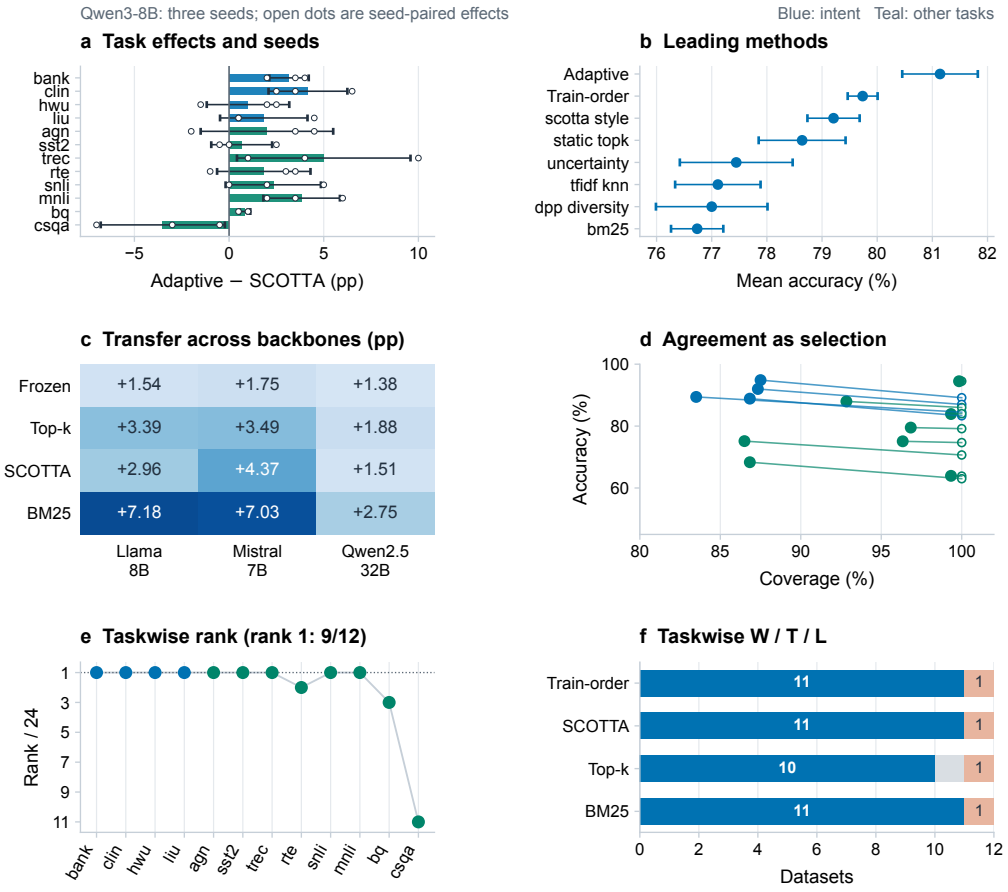

Figure 3: **Six views of the evidence.** (a) Adaptive minus SCOTTA-style; open dots are paired seeds, bars are task means, and whiskers are seed SDs. (b) Leading methods by overall mean. (c) Cross-model gains from the execution ledger underlying Table 2. (d) Full-coverage (open) versus agreement-only (filled) accuracy; lines join the same task, not operating curves. (e) Adaptive's rank among 24 methods. (f) Taskwise wins/ties/losses (blue/gray/orange) from three-seed means.

Table 2: **Cross-model comparison.** Mean accuracy (%) over 12 datasets and three seeds per model (500 training examples, 200 queries, $k = 16$). Pooled values summarize 21,600 predictions. Fixed configurations and single-arm baselines are shown alongside Adaptive; these are system comparisons, not uniformly single-module ablations. Full results and reported paired tests appear in the appendix.

Method	`llama3.1:8b`	`mistral:7b-instruct`	`qwen2.5:32b-instruct`	Pooled
EvoLabel-Adaptive (ours)	**65.58**	**66.71**	**87.62**	**73.31**
fixed: ensemble arbiter	64.18	64.89	86.14	71.74
fixed: dual-view arbiter	64.26	66.64	87.40	72.77
frozen dual-arm (no evolution)	64.04	64.96	86.25	71.75
static_topk	62.19	63.22	85.75	70.39
scotta_style	62.62	62.33	86.11	70.36
topk_sd	61.83	61.44	85.90	69.73
bm25	58.40	59.68	84.88	67.65
random	23.53	23.53	23.53	23.53

7.4 Operator transfer and the observed evolution null

Figure 4 complements the aggregate results with two diagnostics. The error-composition panel separates agreement and disagreement for both task families. Recorded runtime summarizes actual

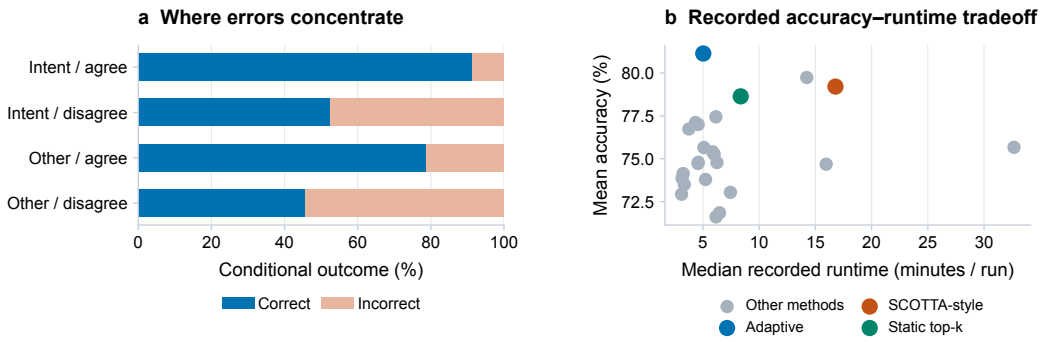

Figure 4: **Diagnostic views.** (a) Correct (blue) and incorrect (light orange) predictions in the six-ballot trace pool, split by task family and by whether the two arms agreed. (b) Main-grid mean accuracy versus median logged runtime across 36 runs per method; runtime is observational, not a controlled speed comparison. The frequency-matched confusion audit is reported numerically in Table 20 rather than repeated as a panel.

executions, but hardware load, caching and execution order are not controlled here; the plot is descriptive and does not establish a hardware-independent efficiency ranking.

The held-out confusion-pair audit discovers pairs on seed 42 and evaluates them on seed 43 with frequency-matched controls (Table 20). The reported 13.6 pp excess error relative to the corpus average disappears under that control (permutation $p \in [0.38, 0.76]$), with oracle precision@5 of 0.10. This limits the evidence for transferable pair selection; it does not show that all possible operators fail. In the evolution-enabled replication runs, the reflector commits no edits. Prediction gains therefore cannot be attributed to executed label updates. Earlier operator variants and extended arbiter tests remain in Appendices A.8 and A.7; they answer distinct questions under different evaluation protocols.

7.5 Interpreting the combined evidence

The benchmark supports a narrower conclusion than the name "self-improving" might suggest. The strongest result concerns prediction: exposing the same fixed pool through complementary retrieval views and resolving their disagreements improves mean accuracy on this suite. The frozen dual-arm, fixed-expert, and adaptive configurations identify where that gain enters the pipeline, but the adaptive route changes a challenger–arbiter pair and remains an end-to-end configuration selected on these datasets. Agreement provides an operational triage signal rather than calibrated confidence: the arms share a model and pool, and abstention trades coverage for accuracy.

The zero-edit trace identifies a failure in this implementation's activation/validation path. It contains no executed treatment contrast from which to estimate rewrite efficacy. A stronger test should pre-register an activating operator, replay each accepted edit against a matched frozen counterfactual, measure collateral errors outside its target pair, and choose the routing rule without the evaluation datasets. Appendix A.1 separates activation, commitment, and causal efficacy formally.

8 Conclusion and Limitations

Complementary prediction views and disagreement arbitration improve the tested classifier, while the tested label-evolution layer makes no committed edits. The conditional decomposition separates this empirical null from a universal impossibility claim. The study is limited by a small, fixed dataset suite, benchmark-calibrated routing, repeated queries across seeds, and a fixed training pool. Adaptive reaches 81.1% across twelve datasets, exceeds the frozen system on three additional backbones, and uses agreement to identify a high-accuracy subset. This supports disagreement-triggered arbitration with full-coverage and cost reporting, but not autonomous taxonomy revision. Unseen-dataset selection and matched counterfactual evaluation of executed edits remain required.

AI USE STATEMENT

Generative-AI assistants were used for manuscript drafting and editing, LaTeX layout, plotting and data-processing code, figure design, consistency checks, and feedback on statistical interpretation and theoretical assumptions. This included assistance revising the statement and exposition of the conditional gain decomposition. The visualizations use recorded experiment outputs rather than synthetic experimental observations. Authors are responsible for verifying the final text, proofs, data provenance, and code before submission.

ETHICS STATEMENT

This study evaluates classification on existing text benchmarks and does not recruit human participants. Benchmark text can contain sensitive or biased content; use and redistribution remain subject to dataset licenses and privacy conditions. Results on these benchmarks do not establish safety for deployment in consequential decision systems.

REPRODUCIBILITY STATEMENT

The appendix specifies the datasets, configurations, supplementary comparisons, and assumptions behind the gain decomposition. Trace-derived figures retain source manifests and reproducible plotting code. Results are based on recorded model executions; re-sampling a served model need not reproduce each prediction bitwise. The anonymous supplementary artifact contains plotting code, the adopted main-grid traces and the per-query prediction records underlying the paired comparisons of Appendix A.4; cross-model results include an adopted-cell ledger. Bootstrap intervals use a fixed RNG seed (20260918) and 20,000 resamples. Historical diagnostics whose exact runner snapshot is unavailable are marked as such in the appendix and do not support the headline comparison.

REFERENCES

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT)*, pp. 92–100, 1998.

Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/aa2a77371374094fe9e0bc1de3f94ed9-Abstract.html.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A Large Annotated Corpus for Learning Natural Language Inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015. URL https://aclanthology.org/D15-1075/.

Tom Brown, Benjamin Mann, Nick Ryder, et al. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, 2020. URL https://aclanthology.org/2020.nlp4convai-1.5/.

Cheng Chen, Yunpeng Zhai, Yifan Zhao, Jinyang Gao, Bolin Ding, and Jia Li. Provoking multimodal few-shot lvlm via exploration-exploitation in-context learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2506.09473.

Haokun Chen, Xu Yang, Yuhang Huang, Zihan Wu, Jing Wang, and Xin Geng. Manipulating the label space for in-context classification. *arXiv preprint arXiv:2312.00351*, 2023. URL https://arxiv.org/abs/2312.00351.

C. K. Chow. An optimum character recognition system using decision functions. *IRE Transactions on Electronic Computers*, EC-6(4):247–254, 1957.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions. In *Proceedings of NAACL-HLT*, 2019. URL https://aclanthology.org/N19-1300/.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1107–1128, 2024. URL https://aclanthology.org/2024.emnlp-main.64/.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, et al. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Chrisantha Fernando, Dylan Sunil Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pp. 13481–13544, 2024. URL https://proceedings.mlr.press/v235/fernando24a.html.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pp. 4878–4887, 2017. URL https://mlanthology.org/neurips/2017/geifman2017neurips-selective/.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A Deep Neural Network with an Integrated Reject Option. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/9156b0f6dfa9bbd18c79cc459ef5d61c-Abstract-Conference.html.

Yinhan He, Wendy Zheng, Song Wang, Zaiyi Zheng, Yushun Dong, Yaochen Zhu, and Jundong Li. Hierarchical demonstration order optimization for many-shot in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/8519ec349e6494297941f8c05ad9420f-Abstract-Conference.html.

Giwon Hong, Emile van Krieken, Edoardo M. Ponti, Nikolay Malkin, and Pasquale Minervini. Mixtures of in-context learners, 2024. URL https://arxiv.org/abs/2411.02830.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2225–2240, Dublin, Ireland, 2022. Association for Computational Linguistics. ACL Anthology 2022.acl-long.158.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. URL https://arxiv.org/abs/2310.06825.

Heinrich Jiang, Been Kim, Melody Y. Guan, and Maya Gupta. To trust or not to trust a classifier. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, pp. 5541–5552, 2018. URL https://mlanthology.org/neurips/2018/jiang2018neurips-trust/.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? On the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021. ACL Anthology 2021.tacl-1.57.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, 2017. URL https://arxiv.org/abs/1703.04730.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://mlanthology.org/iclr/2023/kuhn2023iclr-semantic/.

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

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. An Evaluation Dataset for Intent Classification and Out-of-Scope Prediction. In *Proceedings of EMNLP-IJCNLP*, 2019. URL https://aclanthology.org/D19-1131/.

Itay Levy, Ben Bogin, and Jonathan Berant. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1401–1422, Toronto, Canada, 2023. Association for Computational Linguistics. ACL Anthology 2023.acl-long.78.

Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. URL https://aclanthology.org/C02-1150/.

Fuxiao Liu, Paiheng Xu, Zongxia Li, Yue Feng, and Hyemi Song. Towards understanding in-context learning with contrastive demonstrations, 2024. URL https://arxiv.org/abs/2307.05052.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pp. 100–114, Dublin, Ireland and Online, 2022. Association for Computational Linguistics. ACL Anthology 2022.deelio-1.10.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. Benchmarking natural language understanding services for building conversational agents. In Erik Marchi, Sabato Marco Siniscalchi, Sandro Cumani, Valerio Mario Salerno, and Haizhou Li (eds.), *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction*, volume 714 of *Lecture Notes in Electrical Engineering*, pp. 165–183. Springer, 2021. doi: 10.1007/978-981-15-9323-9_15.

Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022. URL https://aclanthology.org/2022.acl-long.556/.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, Chenlin Zhou, Jiayi Mao, Tianze Xia, Jiafeng Guo, and Shenghua Liu. A survey of context engineering for large language models. *arXiv preprint arXiv:2507.13334*, 2025. URL https://arxiv.org/abs/2507.13334.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 11048–11064, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. ACL Anthology 2022.emnlp-main.759.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M. Waleed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations (ICLR)*, volume 2025, pp. 34433–34448, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/5503a7c69d48a2f86fc00b3dc09de686-Paper-Conference.pdf.

Owen Patterson and Chee Ng. Enhancing contrastive demonstration selection with semantic diversity for robust in-context machine translation, 2025. URL https://arxiv.org/abs/2504.09305.

Keqin Peng, Liang Ding, Yancheng Yuan, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. Revisiting demonstration selection strategies in in-context learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9090–9101, Bangkok, Thailand, 2024. Association for Computational Linguistics. ACL Anthology 2024.acl-long.492; data- and model-dependent demonstration selection.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True Few-Shot Learning with Language Models. In *Advances in Neural Information Processing Systems*, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/5c04925674920eb58467fb52ce4ef728-Abstract.html.

Kiran Purohit, V. Venktesh, Raghuram Devalla, Krishna Mohan Yerragorla, Sourangshu Bhattacharya, and Avishek Anand. EXPLORA: Efficient exemplar subset selection for complex reasoning, 2024. URL https://arxiv.org/abs/2411.03877.

Qwen, An Yang, Baosong Yang, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of EMNLP-IJCNLP*, 2019. URL https://aclanthology.org/D19-1410/.

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

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Selective annotation makes language models better few-shot learners. In *International Conference on Learning Representations (ICLR)*, 2023.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7080–7106, Rabat, Morocco, 2026. Association for Computational Linguistics. ACL Anthology 2026.eacl-long.333.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge. In *Proceedings of NAACL-HLT*, 2019. URL https://aclanthology.org/N19-1421/.

Francesco Tonolini, Nikolaos Aletras, Jordan Massiah, and Gabriella Kazai. Bayesian prompt ensembles: Model uncertainty estimation for black-box large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12229–12272, Bangkok, Thailand, 2024. Association for Computational Linguistics. ACL Anthology 2024.findings-acl.728.

M. S. Vinay, Minh-Hao Van, and Xintao Wu. In-context learning demonstration selection via influence analysis, 2024. URL https://arxiv.org/abs/2402.11750.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 9840–9855, Singapore, 2023a. Association for Computational Linguistics. ACL Anthology 2023.emnlp-main.609; label tokens act as information anchors.

Song Wang, Zihan Chen, Chengshuai Shi, Cong Shen, and Jundong Li. Mixture of demonstrations for in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a0da098e0031f58269efdcba40eedf47-Abstract-Conference.html.

X. Wang and W. Jia. Meta-sel: Efficient demonstration selection for in-context learning via supervised meta-learning, 2026. URL https://arxiv.org/abs/2602.12123.

X. Wang, J. Wu, Y. Yuan, D. Cai, M. Li, and W. Jia. Demonstration selection for in-context learning via reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267, 2025. URL https://mlanthology.org/icml/2025/wang2025icml-demonstration/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of NAACL-HLT*, 2018. URL https://aclanthology.org/N18-1101/.

J. Wu, X. Wang, and W. Jia. Enhancing text annotation through rationale-driven collaborative few-shot prompting. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2025.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1423–1436, Toronto, Canada, 2023. Association for Computational Linguistics. ACL Anthology 2023.acl-long.79; per-sample select-then-rank adaptation.

Pengyu Xu, JingRen Hou, Liping Jing, and Jian Yu. Structured confidence–guided online adaptation for LLM-based multi-label classification. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16671–16686, San Diego, California, USA, 2026. Association for Computational Linguistics. ACL Anthology 2026.findings-acl.822.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Jianfei Zhang, Bei Li, Jun Bai, Rumei Li, Yanmeng Wang, Chenghua Lin, and Wenge Rong. Selecting demonstrations for many-shot in-context learning via gradient matching. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025a. URL https://arxiv.org/abs/2506.04579.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models. In *International Conference on Learning Representations (ICLR)*, volume 2026, pp. 86069–86100, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8a94ff6f922d995d7d3f4ebf4143e442-Abstract-Conference.html.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, pp. 649–657, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html.

Ziniu Zhang, Zhenshuo Zhang, Dongyue Li, Lu Wang, Jennifer Dy, and Hongyang R. Zhang. Linear-time demonstration selection for in-context learning via gradient estimation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 16459–16477, Suzhou, China, 2025b. Association for Computational Linguistics. ACL Anthology 2025.emnlp-main.832.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *Proceedings of the 38th International Conference on Machine Learning*, 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations (ICLR)*, volume 2024, pp. 19426–19454, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/54dd9e0cff6d9214e20d97eb2a3bae49-Paper-Conference.pdf.

Yucheng Zhou, Xiang Li, Qianning Wang, and Jianbing Shen. Visual in-context learning for large vision-language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL https://arxiv.org/abs/2402.11574.

A Supplementary Analysis and Reproducibility

A.1 Proof and multi-round scope

Valid presentation states. A state $L = (\mathcal{Z}, \pi, d)$ is valid when every displayed entry maps to a nonempty subset of $\mathcal{Y}$, the image of π covers $\mathcal{Y}$, and the canonical decoder returns an element of $\mathcal{Y}$. These conditions make presentation edits type-safe while leaving the evaluation inventory immutable. Merge modifies $\mathcal{Z}$ and π; Describe modifies d. Neither operation, by definition, changes a target y, an example in P, or model parameters. State validity is weaker than predictive validity: a well-formed edit can still change a correct output to an incorrect one.

Single-step identity. For any two fixed systems s_0, s_1 and measurable subset D, the law of total probability gives

$$\mathrm{acc}(s_1) - \mathrm{acc}(s_0) = \Pr[D]\Delta_D + \Pr[D^c]\Delta_{D^c}.$$

Here $\Delta_B = \Pr[s_1 = y \mid B] - \Pr[s_0 = y \mid B]$. Assumption 5.1 sets $\Delta_{D^c} = 0$, proving Proposition 5.2. Since conditional accuracy cannot exceed one,

$$\Delta_D \le 1 - \Pr[s_0 = y \mid D],$$

which gives the stated oracle ceiling. The equality and ceiling concern a specified pair of systems and a specified subset. They do not imply that the implemented gate enforces the assumption.

Multi-round endpoint accounting. For a trajectory $s_0, \ldots, s_T$, define actual prediction-change sets $A_t = \{x : s_{t+1}(x) \neq s_t(x)\}$ and $U = \bigcup_{t<T} A_t$. For every $x \notin U$, the endpoint predictions are identical, hence

$$\mathrm{acc}(s_T) - \mathrm{acc}(s_0) = \Pr[U]\{\Pr[s_T = y \mid U] - \Pr[s_0 = y \mid U]\}.$$

Therefore the endpoint upper bound is $\Pr[U]\{1 - \Pr[s_0 = y \mid U]\}$. This formulation remains exact when change sets overlap, avoiding a sum of per-round effects that would double-count queries. Replacing U by a baseline disagreement set is valid only after establishing the corresponding containment or invariance. Similar disagreement rates across rounds are not sufficient.

Three quantities that must remain separate. An evolution study has (i) *activation*, whether the reflector proposes an operation; (ii) *commitment*, whether validation changes L_t; and (iii) *causal efficacy*, the paired effect of the committed state against a frozen counterfactual. Zero committed operations identifies a failure before stage (iii): it is evidence about this activation/validation path, but contains no executed treatment contrast from which to estimate rewrite efficacy.

A.2 Empirical factors and their interpretation

The earlier seven-variant diagnostic pool reports 1,505 disagreements among 11,200 predictions (13.44%), with a reported conditional contrast of $50.06 - 47.84 = 2.22$ pp. Their product is about 0.30 pp. This observed contrast is not an upper bound for all future arbiters or label rewrites. Accuracy near 50% is also not, by itself, proof of chance performance: neither candidate coverage nor a balanced binary decision problem is guaranteed. The corresponding perfect-correction ceiling is approximately 7.0 pp under the stated invariance. A fixed evidence pool can still yield different decisions under different presentations; no Bayes-error impossibility result follows from the present data. Figure 5 illustrates only the algebraic dependence. A useful future study would estimate paired uncertainty and execute edits on independently chosen tasks before treating small estimated efficacy as a deployment criterion.

The two factors answer different design questions. The affected-query mass is a reach term: even a perfect local repair cannot improve examples whose output does not change. The conditional contrast is an efficacy term: reaching many queries is undesirable if their conditional accuracy falls. Their product is useful for pre-registering the minimum detectable effect, but only after the affected set is defined independently of the outcomes being summarized. In particular, selecting a confusion pair and evaluating it on the same stream would inflate both reach and apparent efficacy; the held-out and frequency-matched audit in Section A.17 addresses part of that selection problem.

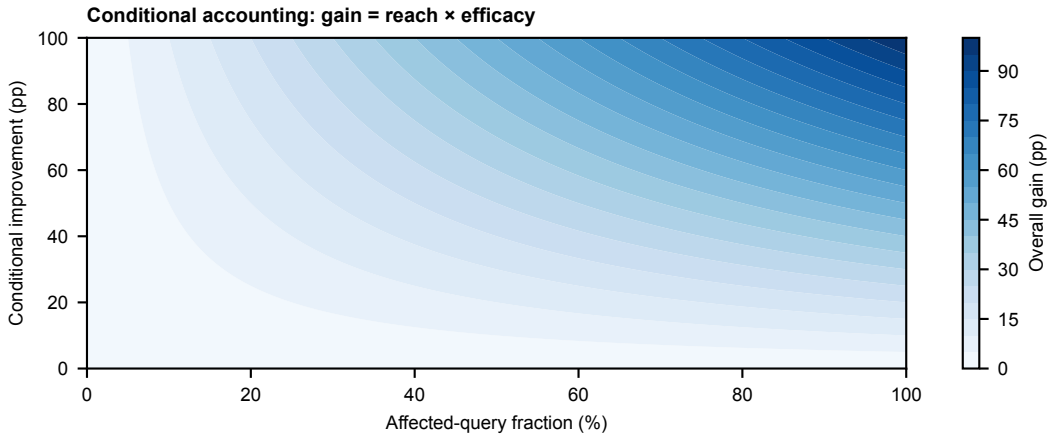

Figure 5: **Conditional gain accounting.** The displayed surface is the mathematical product of affected-query fraction and conditional accuracy improvement; it is not an empirical estimate or a universal bound on rewriting.

The approximately 50% conditional accuracies in the archived diagnostic do not establish random guessing. Some disagreements contain neither correct arm, some may involve more than two effective candidates, and the conditional class distribution need not be balanced. For the same reason, an oracle that chooses a correct arm when available is a restricted decision oracle rather than a Bayes-optimal classifier. We retain the numerical decomposition as descriptive accounting and avoid the stronger information-theoretic interpretation used in an earlier draft.

A.3 Protocols and statistical units

The headline Qwen3-8B grid contains 24 methods, 12 datasets and three seeds, with 200 queries per run. Table 1 and its SD companion are recomputed from the same 864 JSON records; each file contains 200 predictions and its recorded accuracy is checked against the correctness flags. The figure source manifest retains the file hash, method, dataset, seed and accuracy. Seed SDs describe repeat variability, not confidence intervals for transfer to new datasets. Figure 3a uses paired seed-level differences. Panel b first averages datasets within a seed, then reports the SD of those three means. The heatmap in panel c is a difference of published summary means, not a new test of significance.

Pooled McNemar results in Table 15 are retained as originally reported. Repeated queries and shared datasets mean that a nominal count of 21,600 predictions is not 21,600 independent task-level observations. Those p-values are not cluster-adjusted evidence of new-dataset generalization, and the multiple comparisons are not presented as a family-wise controlled ranking. Dataset-level uncertainty and independent routing calibration remain important limitations. The many-class subset comprises only four intent tasks.

The six-ballot motivation and selective plots use 36 explicitly identified `evolabel_v6` records. All their coverage gates are open and operation counts are zero. These executions differ from earlier reported arbiter summaries; protocol labels distinguish the tables rather than treating them as identical replications. Raw-record checks also find zero operations in all 36 train-order and all 36 adaptive records in the headline grid; external baselines do not have a reflector, so "864 zero-edit reflectors" would be misleading.

A.4 Inference for the headline comparison

The headline comparison pairs EvoLabel-Adaptive against the SCOTTA-style comparator on the main grid (12 tasks, three seeds, 200 queries). Recomputing both means from the recorded per-query correctness flags reproduces the published values (81.14% and 79.21%, a +1.93 pp gap), which validates the extraction used below.

Table 3: **Retrieval arms.** The arms differ only in how in-context evidence is constructed; the pool, model, decoder and k are shared.

Arm	Signal	Selection rule
F (fast)	SBERT cosine	top-k embedding neighbours of x under L_0
R (diverse)	SBERT cosine with MMR	promotes semantic diversity among the k slots under L_t
M (lexical)	BM25	top-k lexical matches of x

Table 4: **Arbiters.** A_b builds b independent demonstration slates, takes one canonical vote per slate and returns the majority label; ties are resolved by p_F. Arbitration pools evidence-construction choices, not resampled generations.

Arbiter	b	Slate construction
A_6	6	six candidate-focused query-relevance slates
A_{12}	12	eight semantic plus four lexical slates

Primary analysis: dataset as the unit. Following Appendix A.3, each task contributes one difference of three-seed means, giving twelve paired differences. A dataset-clustered bootstrap (20,000 resamples of the twelve tasks) yields a 95% interval of [+0.65, +3.03] pp, and a paired Wilcoxon signed-rank test over the twelve tasks gives $W = 9.0$, $p = 0.016$, with Adaptive ahead on eleven of twelve tasks. The sole exception is CommonsenseQA (−3.5 pp); per-task differences range from −3.5 to +9.8 pp, which is why we report an interval and a win count instead of a single margin.

Supporting query-level test and its exclusions. A paired query-level McNemar test requires both methods to be evaluated on identical queries. Auditing the traces shows this holds for ten of the twelve tasks. For SNLI and TREC, some runs were executed against non-identical query samples (9/200 and 84/200 overlap at seed 42), so both tasks are excluded from every paired query-level analysis. On the remaining 6,000 paired queries, Adaptive is correct where the comparator is wrong on 311 queries versus 216 in the opposite direction (McNemar $\chi^2 = 16.8$, $p = 4.2 \times 10^{-5}$). This p-value is reported as supporting evidence only: repeated queries and shared tasks are not independent observations (Appendix A.3), so the dataset-level interval above is the primary inferential statement. The two affected tasks remain in the dataset-level means, where they contribute unbiased per-task accuracies.

Intent subset. On the four intent tasks the gap is +2.54 pp (bootstrap interval [+1.42, +3.67] pp, 4/4 tasks). A Wilcoxon test on four units cannot reach conventional significance—the minimum attainable two-sided p at $n = 4$ is 0.125—so we report no p-value for that subset and rely on the interval.

A.5 Method specification: retrieval arms, arbitration, and reflector

This subsection specifies the components summarized in Section 4, keeping the body within the page limit while leaving every configuration reproducible. All variants share the pool P, the backbone h_θ, the prompt template, $k = 16$, the canonical decoder ρ_L and the canonical sorted label order. The degrees of freedom are exactly (i) which challenger view is queried, (ii) how the arbiter aggregates ballots, and (iii) whether the reflector is enabled. Table 3 defines the retrieval arms, Table 4 the arbiters, Table 5 the resulting dataset-level routing, and Table 6 the reflector constraints.

Arbiter conventions. Every ballot shares the pool P, the backbone h_θ and the decoder ρ_L, so the arbiter pools *evidence-construction* choices rather than resampled generations from one fixed context. It is therefore distinct from self-consistency sampling, and we claim no conditional independence between ballots. Each slate contributes exactly one vote, so aggregation is an unweighted majority rather than a confidence-weighted rule: this keeps every ballot's construction cost identical and avoids fitting ballot weights to the evaluation data. Ties are resolved toward p_F, the prediction that would have been returned had no second view been queried, which makes the rule deterministic and

Table 5: **Dataset-level routing.** The gate g is computed once per dataset from the pool (Eq. 3); the routing pair is fixed before test inference and never uses query-level labels or outcomes.

Condition	Challenger S	Arbiter A
$g = 0$	none (arm F only)	none
$g = 1$, $\lvert\mathcal{Y}\rvert \geq 20$	R	A_6
$g = 1$, $\lvert\mathcal{Y}\rvert < 20$	M	A_{12}

Table 6: **Reflector constraints** (label-free configuration).

Quantity	Value / rule
Window	$\lvert W_t\rvert \leq 64$ most recent queries
Pair support	directed counts $N_t(a, b)$ over (p_F, p_S); a candidate becomes eligible at support ≥ 4
Operation budget	at most 8 operations per run
Operators	MERGE: replace two entries by their union; DESCRIBE: edit d_t from prototypical surface forms
Validity check	$q(L_t, o_t) \in \{0, 1\}$ tests type and state consistency; when it is 0 the state is unchanged ($L_{t+1} = L_t$)

reproducible and biases arbitration toward the cheaper single-arm outcome. Both conventions are fixed before inference and are not chosen per query.

Call accounting. The prediction-path call count per query is

$$K(x) = 1 + g\{1 + D(x)b_A\}, \tag{7}$$

with $D(x)$ the disagreement indicator of Eq. 4 and $b_A \in \{6, 12\}$ the ballot count: one call with a closed gate, two on agreement, and $2 + b_A$ on disagreement (eight for A_6, fourteen for A_{12}). Reflection adds round-level overhead only when the layer is enabled and never adds a second prediction-path call for the scored query, so arbiter cost is proportional to the disagreement rate—which is why disagreement rates are reported alongside accuracy throughout. Observed runtime additionally reflects serving load, caching and execution order and is therefore reported as descriptive evidence rather than a hardware-normalized complexity result.

State boundaries. The scored prediction for x_t is finalized before any feedback from x_t reaches the reflector, so an accepted operation can affect only later retrieval and presentation. This ordering is what makes frozen and evolution-enabled runs comparable at the query level, and it prevents a same-query update from using its own outcome retroactively.

A.6 Implementation details: hyperparameters and prompt

All 864 runs share one protocol. Table 7 summarizes the main settings; the classifier prompt (identical for both arms and all baselines, modulo the demonstration set and the allowed-label list) is given verbatim below. The label inventory is presented in a *canonical sorted order shared across every method*—we measured that presentation order alone shifts accuracy by up to 7.5 pp on banking77, so fixing it removes a confound that would otherwise dwarf the method effect.

Classifier prompt (line-wrapped for display).

```
You are a strict text classifier.
Choose exactly one LABEL_NUMBER from the allowed list.
Allowed labels:
0 = <label_0>
1 = <label_1>
...
Examples:
TEXT: <demo text>
LABEL_NUMBER: <demo label id>
```

Table 7: **Shared protocol and method-specific settings.**

Parameter	Value
Base model	`qwen3:8b` (open-weight, served via local Ollama)
Decoding	temperature 0.0, `num_predict` 32, stop at newline, thinking disabled
Demonstrations per arm (k)	16
Training / test queries per cell	500 / 200
Seeds	42, 43, 44
Retriever	`all-MiniLM-L6-v2` (Sentence-Transformers); pool = training set
Arm F (frozen)	Top-k retrieval over the full label inventory
Arm R	Diversity-based (MMR-style) retrieval over the (evolved) label space; allowed labels = labels present in the retrieved demos
Coverage gate	pool leave-one-out any-hit@$k \geq 0.70 \Rightarrow$ gate open
Reflector (evolution layer)	`max_ops`= 8, `window`= 64, `min_support`= 4; symmetric confusion gate (both $a \to b$ and $b \to a$ required)
Arbiter (ballot-ensemble, deployed)	6-ballot ensemble over query-relevance ranking slices; stronger arm presented first; majority vote
Label presentation	canonical sorted label order, shared across all methods
Gold probe (audits only)	0%/10%/100% of round queries; reflector-only, never used to alter predictions or demonstrations

Table 8: **Evaluation tasks.** Nominal labels describe the source task; pool labels report the labels represented after the fixed 500-example sample. Accuracy is the common metric.

Dataset	Prediction target	Domain	Source	Nominal	Pool
Banking77	customer intent	banking queries	(Casanueva et al., 2020)	77	76
CLINC150	assistant intent	virtual assistants	(Larson et al., 2019)	150	144–147
HWU64	assistant intent	home/robotics	(Liu et al., 2021)	64	64
Liu54	user intent	multi-domain queries	(Liu et al., 2021)	54	50
AG News	news topic	news headlines	(Zhang et al., 2015)	4	4
SST-2	sentiment	movie reviews	(Wang et al., 2019)	2	2
TREC	coarse question type	factoid questions	(Li & Roth, 2002)	6	6
RTE	textual entailment	mixed-domain pairs	(Wang et al., 2019)	2	2
SNLI	NLI relation	image captions	(Bowman et al., 2015)	3	3
MNLI	NLI relation	multi-genre pairs	(Williams et al., 2018)	3	3
BoolQ	yes/no answer	Wikipedia questions	(Clark et al., 2019)	2	2
CommonsenseQA	answer option	commonsense questions	(Talmor et al., 2019)	5	5

```
...
TEXT: <query text>
Return only one integer.
Do not output the label name, explanation, punctuation,
or repeated text.
LABEL_NUMBER:
```

The candidate set in `Allowed labels` is the full inventory for Arm F and the subset retrieved for Arm R; the `Examples` block is the k=16 demonstrations. Predictions are extracted as the first integer matching a valid label id.

A.7 Complete benchmark and cross-model tables

Task construction. The four intent datasets test retrieval against a large candidate inventory and fine-grained semantic overlap. The other eight cover topic classification, sentiment, question type, textual entailment, and multiple-choice reasoning. Each run samples a 500-example pool and a disjoint 200-query evaluation stream under its recorded seed. Nominal source labels can be absent from a finite pool; Table 8 therefore reports both nominal and represented label counts. The many-class/few-class split is simultaneously a task-family split: all four many-class tasks are intent datasets and every other task has at most six labels. Any explanation based on cardinality alone is consequently observational.

Controlled baseline interface. Every selector is fit to the same pool and must return exactly $k = 16$ demonstrations for the common classifier prompt. The wrapper converts pool records to the selector interface, converts selected records back to the shared prompt format, and leaves the LLM call unchanged. This controls the inference backbone and presentation format while varying selection. A method identifier ending in `proxy`, `style`, or supplied through the standardized catalog

Table 9: **Baseline families in the 24-method grid.** Every method uses the same pool, prompt, k, backbone, decoding, and test queries. "Catalog adapter" denotes the locally standardized selector implementation; it should not be interpreted as an exact reproduction of every original paper.

Family	Method identifiers	Controlled difference
Dual-arm systems	`evolabel_v_adaptive`, `evolabel_v2`	routed challenger/settler or train-order settlement
Online/uncertainty	`scotta_style`, `uncertainty`	online selection proxy or uncertainty-prioritized selection
Direct retrieval	`static_topk`, `bm25`, `tfidf_knn`, `label_balanced`, `influence_style`	semantic, lexical, balanced, or influence-style scores
Diversity/prototype	`dpp_diversity`, `kmeans_prototype`, `coreset`, `gradmatch_many_shot`, `diverse_cone`, `topk_cone`	coverage, prototypes, coreset, or gradient/diversity proxies
Catalog adapters	`vicl`, `vlm_e2e`, `contrastive_icl`, `moicl`, `hdoo`, `mod`, `explora`, `inficl`, `vote_k_proxy`	standardized adapters run through the common inference harness

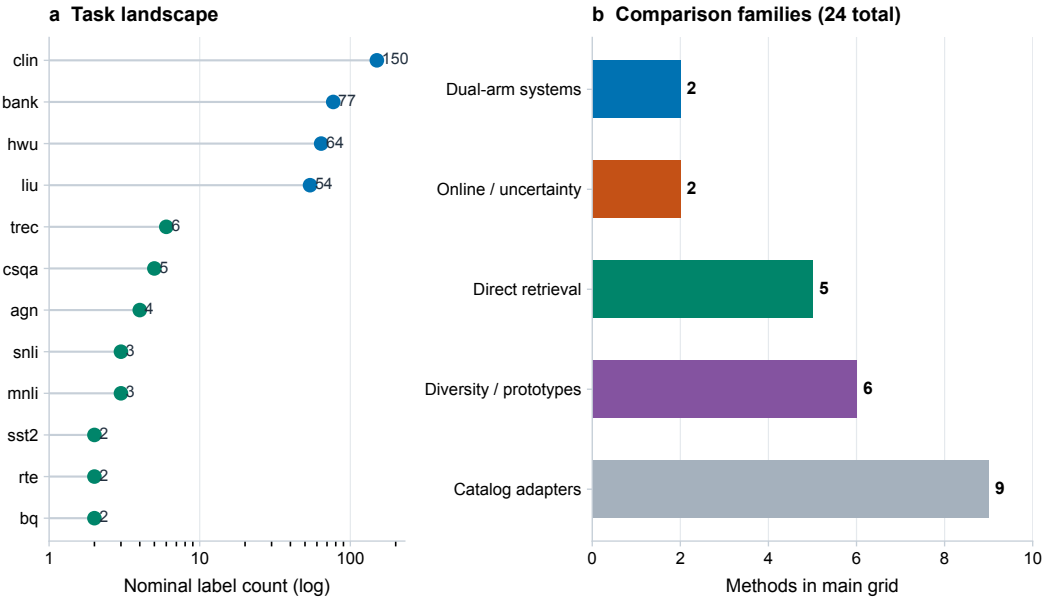

Figure 6: **Benchmark composition.** (a) The suite spans two separated cardinality regimes; task family and cardinality are consequently confounded. (b) The 24 methods grouped by the controlled selection signal. Blue and teal denote the same task families used in the main figures.

names a local controlled implementation. Results establish performance of that implementation in this harness; they are not a claim that all training, indexing, or tuning choices of the original named method were reproduced.

Comparator roles. Static top-k, BM25 and TF–IDF anchor semantic and lexical retrieval. Label-balanced, uncertainty, influence, prototype, DPP and coreset variants probe common selection objectives. SCOTTA-style is the closest online selection-side comparator, while the frozen/train-order dual-arm system is the direct reference for isolating arbitration or executed rewrite effects. Named catalog adapters broaden the empirical comparison but are not used to define theoretical assumptions. The complete table is retained because selecting only the strongest rows after seeing the outcomes would conceal method-level heterogeneity.

Baseline descriptions. Table 9 groups the 24-method grid into families; we now introduce each method individually and cite its originating work. Every method runs through the controlled harness of Appendix A.7 (identical pool, prompt, k, backbone, decoding and test queries), so any accuracy difference reflects the selection logic rather than the setup. The names in `small caps` match the identifiers in the table.

Direct retrieval and simple baselines

Static top-k (Reimers & Gurevych, 2019) embeds queries and candidates with a fixed Sentence-BERT encoder and retrieves the k nearest candidates by cosine similarity, fixing the demonstration pool once; it is the no-evolution reference for every EvoLabel variant.

BM25 (Robertson & Zaragoza, 2009) ranks candidates by the probabilistic relevance of overlapping terms, providing an unsupervised, embedding-free counterpoint to semantic retrieval.

TF–IDF kNN (Liu et al., 2022) is the simplest vector-space retriever: it embeds candidates with TF–IDF features and selects the k nearest neighbours of the query.

Label-balanced (Levy et al., 2023) enforces coverage across the label space so that the chosen pool is not dominated by frequent labels; it tests whether balancing alone reaches the same accuracy as learned selection.

Influence-style (Koh & Liang, 2017) scores each candidate by its estimated influence on the inference outcome using influence functions, prioritising examples that most move the model's prediction; it is the only baseline that explicitly models example–prediction interactions.

Random (Brown et al., 2020) draws k candidates uniformly at random and serves as the floor that every learned method must beat.

Online and uncertainty-adaptive selection

Uncertainty (Kuhn et al., 2023) selects candidates that maximise the model's predictive uncertainty (entropy or semantic-uncertainty estimates), targeting examples on which the model is least confident.

SCOTTA-style (Xu et al., 2026) is an online selection proxy that, for each query, first estimates which labels are easy versus hard from a validation split and then adapts the demonstration mix accordingly; it is the strongest published adaptive baseline we compare against.

Diversity and prototype selection

DPP diversity (Kulesza & Taskar, 2012) uses a determinantal point process to favour diverse, low-redundancy demonstration sets by maximising the determinant of a similarity kernel over the chosen candidates.

K-means prototype (Lloyd, 1982) partitions the candidate pool with k-means and keeps one representative per cluster, trading fine-grained similarity for compact, class-spread coverage.

Coreset (Borsos et al., 2020) builds a small weighted summary of the pool intended to be sufficient for inference, selecting representatives that minimise redundancy.

Grad-Match (many-shot) (Zhang et al., 2025a) extends gradient-matching subset selection from few-shot to many-shot ICL, choosing demonstrations whose gradients best approximate the full-pool gradient; it is the only baseline that uses gradient alignment at the example level.

TopK+ConE (Peng et al., 2024) pairs a top-k retriever with a contrastive embedding objective to separate relevant from irrelevant candidates; it is the direct ancestor of DiverseConE.

DiverseConE (Patterson & Ng, 2025) augments contrastive selection with a semantic-diversity term so that the chosen demonstrations are both relevant and mutually distinct.

Standardized catalog adapters Each adapter below is the locally standardized selector implementation of a recent ICL-selection method, run through our common inference harness; it should be read as a faithful-in-spirit reproduction rather than an exact replication.

V-ICL (Zhou et al., 2024) a visual in-context learning adapter that conditions the model on a visual demonstration template; included to test whether vision-oriented ICL adapters transfer to text classification.

VLM end-to-end (Chen et al., 2025) an exploration–exploitation ICL strategy from the multimodal literature (Chen et al., CVPR 2025) that balances trying novel demonstrations against exploiting known-good ones; we run its selection head through our text harness.

Contrastive-ICL (Liu et al., 2024) adopts the recipe that ranks candidates by their contrastive agreement with the query, studying how positive/negative demonstrations change ICL behaviour.

MoICL (Hong et al., 2024) a mixture-of-in-context-learners adapter that ensembles several demonstration policies; it tests whether mixing selection strategies helps.

HIDO (He et al., 2025) Hierarchical Demonstration Order Optimization for many-shot ICL (He et al., NeurIPS 2025), which orders demonstrations hierarchically to stabilise long contexts; included as a many-shot ordering baseline.

MoD (Wang et al., 2024) Mixture-of-Demonstrations (Wang et al., NeurIPS 2024), which learns to combine heterogeneous demonstration sets; a representative learnable-mixture baseline.

EXPLORA (Purohit et al., 2024) an efficient exemplar-subset selection method for complex reasoning (Purohit et al., 2024) that scores candidates for reasoning tasks; we evaluate it on classification to test transfer.

InfICL (Vinay et al., 2024) selects demonstrations via influence analysis (Vinay et al., 2024), estimating each candidate's effect on the query prediction; conceptually close to Influence-style but derived from a dedicated influence-estimation objective.

Vote-k proxy (Su et al., 2023) a selective-annotation baseline (Su et al., ICLR 2023) that chooses a diverse, high-coverage seed set by kernel-based voting; it is also the unsupervised pool-initiation strategy we adopt for EvoLabel's candidate pool.

Dual-arm systems (proposed) `evolabel_v_adaptive` and `evolabel_v2` are our proposed and compared dual-arm systems; their module-level definitions are given in the Variant definitions in Appendix A.15.

A.8 Earlier evolution and arbiter studies

These tables retain the earlier experimental records, including the four-task, two-seed evolution study and the extended arbiter comparison. Their means are not interchangeable with the trace-derived three-seed main grid. Some earlier label-rewrite implementations are absent from the current code snapshot; reconstruction of those exact runs is therefore a reproducibility limitation. The frozen dual-arm is the relevant comparator for executed rewrite effects. The adaptive system changes the selected retrieval view and arbiter together; not every variant is a single-module ablation.

A.9 Held-out operator audit

Confusion pairs are discovered on seed-42 streams and assessed on seed-43 streams against frequency-matched controls. Table 20 retains the reported audit: the corpus-average contrast does not survive this control. The conclusion concerns this selection procedure and these streams; it does not identify the Bayes error or rule out all label-space edits.

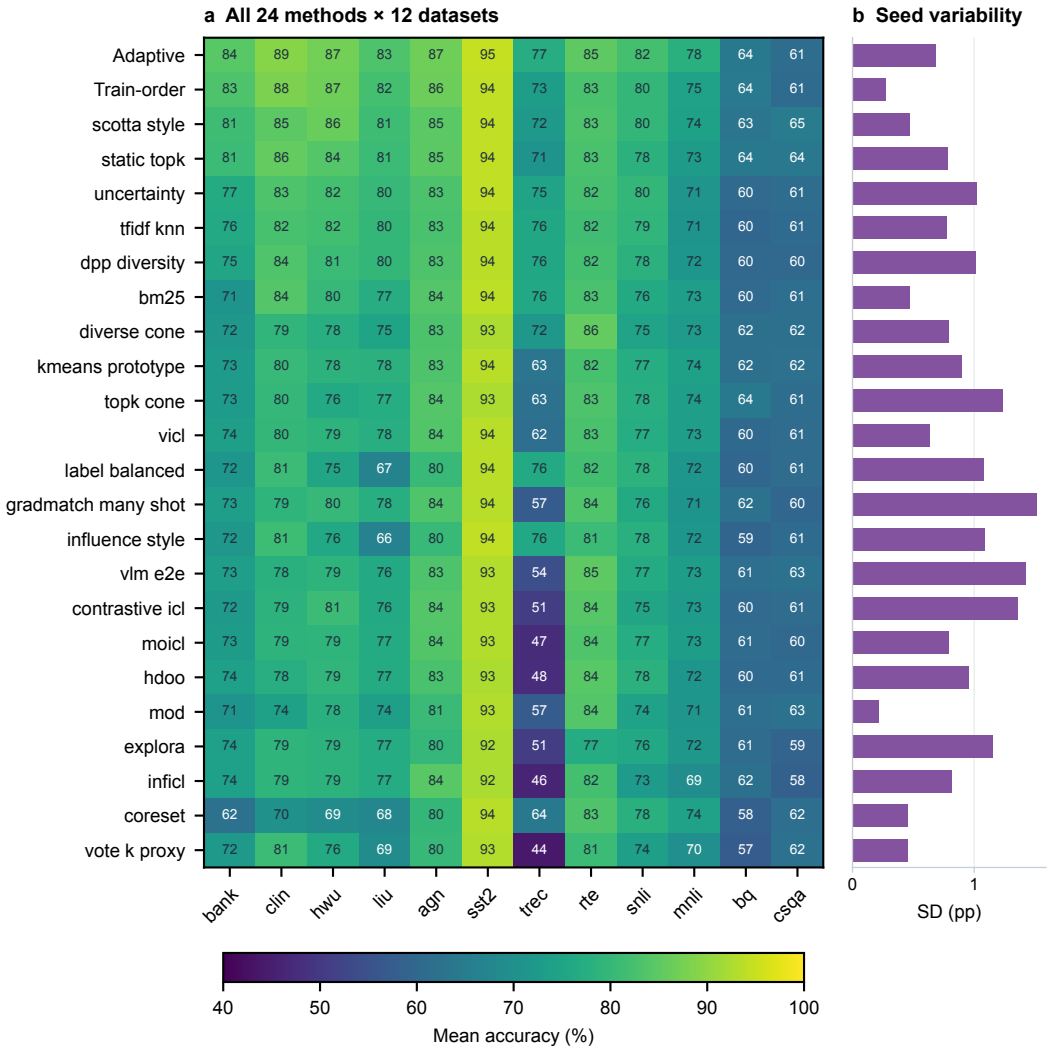

Figure 7: **Complete main-grid landscape.** (a) All 24 methods and 12 tasks, ordered by overall mean accuracy; cell labels round to integers for display. (b) SD of each method's dataset-averaged accuracy across three seeds. Tables 1 and 10 retain decimal precision.

Table 10: Sample standard deviation (percentage points) across three seeds for Table 1. The final column is the SD of each seed's 12-dataset mean, not the mean of cell SDs.

Method	bank	clin	hwu	liu	agn	sst2	trec	rte	snli	mnli	bq	csqa	Mean SD
EvoLabel-Adaptive	1.8	1.8	2.9	1.3	2.5	1.3	3.8	1.3	1.5	2.3	3.5	2.1	0.7
Train-order dual-arm	3.3	1.2	2.8	1.8	2.0	0.8	0.8	1.0	0.3	1.0	1.3	1.6	0.3
scotta_style	1.0	0.3	1.0	1.3	4.8	1.8	3.7	1.8	1.2	0.3	3.3	1.6	0.5
static_topk	2.8	0.5	2.5	1.0	1.8	1.3	4.9	1.3	0.8	5.3	1.5	1.5	0.8
uncertainty	1.6	1.8	3.5	1.3	1.0	0.8	5.5	2.9	2.5	1.0	3.5	2.1	1.0
tfidf_knn	2.3	1.5	4.2	1.5	0.3	0.8	6.0	2.0	3.5	0.3	3.7	2.0	0.8
dpp_diversity	1.3	1.5	3.5	2.0	1.3	1.0	7.1	2.2	3.8	1.7	3.9	2.0	1.0
bm25	2.1	1.0	2.6	1.0	1.9	1.8	4.4	1.3	2.0	0.3	3.8	1.9	0.5
diverse_cone	3.0	2.8	1.7	2.6	1.3	1.5	7.5	0.5	2.8	3.0	1.8	1.3	0.8
kmeans_prototype	2.4	0.9	3.8	2.8	2.3	1.0	4.6	1.3	1.6	1.8	7.5	2.5	0.9
topk_cone	3.1	1.7	1.6	1.3	1.9	1.3	13.6	1.6	3.0	4.4	2.6	4.3	1.2
vicl	4.0	0.8	1.0	0.8	1.9	0.9	7.1	1.9	1.5	1.5	2.8	4.1	0.6
label_balanced	5.6	0.8	3.4	0.3	1.3	0.3	6.4	2.0	3.5	1.0	3.3	2.0	1.1
gradmatch_many_shot	3.4	0.9	1.8	0.5	2.3	1.0	9.1	1.5	2.8	1.3	2.3	3.0	1.5
influence_style	5.7	1.2	3.5	1.2	1.3	0.8	6.6	1.8	3.9	0.5	3.3	2.1	1.1
vlm_e2e	4.5	0.3	2.5	3.1	0.8	1.0	10.8	1.3	3.0	2.8	2.8	2.8	1.4
contrastive_icl	5.4	1.0	1.5	1.8	1.3	1.3	9.3	1.0	2.6	2.8	3.1	1.7	1.4
moicl	3.8	0.3	1.8	1.0	2.5	1.5	0.6	2.2	1.0	2.0	2.5	4.4	0.8
hdoo	3.3	1.0	2.8	1.2	1.3	1.2	1.5	1.0	3.6	1.3	2.8	1.8	1.0
mod	2.2	1.6	2.3	0.6	2.3	0.8	9.1	0.5	2.6	1.6	6.4	2.9	0.2
explora	5.5	1.3	2.5	1.8	0.5	0.9	7.5	1.3	5.3	0.8	2.8	2.5	1.2
inficl	4.3	0.5	2.4	1.3	1.4	0.6	1.5	1.8	5.7	3.3	1.0	1.4	0.8
coreset	3.2	2.0	3.5	3.6	2.5	2.0	6.7	1.3	3.1	1.0	4.2	1.8	0.5
vote_k_proxy	3.3	2.9	4.4	0.6	2.1	0.0	7.7	2.5	2.5	2.5	2.8	2.3	0.5

Table 11: **Multi-model robustness on `llama3.1 8b` (17 methods × 12 datasets × 3 seeds).** Mean accuracy (%) over 3 seeds; best per column in bold, second underlined. Mirrors the main-study protocol (k=16, 500 train / 200 test). EvoLabel-Adaptive (ours) is the proposed adaptive router; its per-dataset breakdown is Table 14.

Method	agn	bank.	bq	clin.	csqa	hwu	liu	mnli	rte	snli	sst2	trec	mean
EvoLabel-Adaptive (ours)	76.3	77.5	56.0	**85.8**	22.8	**85.2**	**80.2**	45.5	62.2	43.3	83.5	68.7	**65.6**
EvoLabel (original evolution)	75.0	71.8	53.5	83.8	20.4	82.8	74.1	35.8	48.2	32.4	80.4	46.9	58.8
EvoLabel Train-order ballot	75.7	73.2	58.0	83.9	23.0	82.2	79.8	46.7	60.8	42.3	80.5	65.5	64.3
BM25 third-arm vote	74.2	70.2	57.3	81.2	22.8	80.6	77.8	45.8	61.0	42.8	81.5	68.0	63.6
fixed: query-relevant arbiter	**77.3**	75.5	57.3	85.0	23.1	84.4	79.0	45.8	60.6	42.1	80.5	67.2	64.8
query-relevant arbiter + swap ballot	75.5	74.0	57.2	84.8	23.2	84.2	79.8	45.3	61.8	42.4	81.2	68.3	64.8
fixed: ensemble arbiter	67.0	**79.2**	57.0	85.8	22.1	84.8	80.0	46.2	61.0	42.6	80.8	63.0	64.1
dual-view + lexical voting (ensemble)	69.0	63.2	56.2	79.2	25.1	78.8	77.8	43.3	60.7	44.2	82.3	**73.3**	62.8
dual-view + lexical voting (3-way)	77.0	64.2	**58.3**	73.3	23.3	72.5	73.5	47.5	62.0	44.8	**83.8**	68.8	62.4
fixed: dual-view arbiter	76.7	68.7	56.7	81.0	23.7	80.2	77.5	47.2	**63.6**	43.2	83.3	69.3	64.2
MMR-diverse arms + ensemble arbiter	74.7	70.8	55.7	80.8	21.8	81.7	77.7	44.3	**63.6**	44.0	82.5	68.3	63.8
scotta_style	75.8	67.5	56.2	75.8	22.0	79.2	75.3	44.7	62.2	**45.1**	79.5	68.0	62.6
static_topk	75.7	67.2	57.7	74.8	22.5	77.4	74.3	46.3	60.3	42.7	80.8	67.2	62.2
topk_sd	71.3	64.7	57.5	76.0	21.4	74.4	72.7	**49.7**	61.9	42.4	80.3	69.3	61.8
bm25	67.4	51.9	55.9	67.3	**26.3**	67.8	71.2	43.3	57.5	44.0	78.8	69.5	58.4
frozen dual-arm (no evolution)	76.2	72.0	56.8	83.2	22.8	83.1	79.2	46.2	60.7	42.3	80.9	66.2	64.1
random	24.0	0.8	52.3	0.8	20.0	1.7	3.0	31.8	47.2	31.7	51.0	18.0	23.5

A.10 Selective prediction in earlier protocols

Table 21 uses the earlier seven-variant two-seed pool. Table 22 retains a separately recorded matched-coverage comparison. Neither supplies the values in the new six-ballot figures, whose source records are identified separately. Disagreement costs no additional inference once the two predictions exist, but obtaining the second prediction has a cost relative to a single-arm system. Abstention also reduces coverage; accuracy conditional on acceptance must not be presented as full-coverage accuracy.

A.11 Design constants, calibration, and robustness

Every fixed quantity in Section 4 is listed in Table 23 together with its role and its status as a degree of freedom. The design principle throughout is that a constant is fixed *before* test inference, is computed from the retrieval pool alone—never from test labels or test-time model outputs—and, wherever it belongs to the evaluation protocol, is shared identically by all compared methods.

Table 12: **Multi-model robustness on `mistral 7b-instruct` (17 methods × 12 datasets × 3 seeds).** Mean accuracy (%) over 3 seeds; best per column in bold, second underlined. Mirrors the main-study protocol (k=16, 500 train / 200 test). EvoLabel-Adaptive (ours) is the proposed adaptive router; its per-dataset breakdown is Table 14.

Method	agn	bank.	bq	clin.	csqa	hwu	liu	mnli	rte	snli	sst2	trec	mean
EvoLabel-Adaptive (ours)	74.3	80.8	62.2	**84.8**	**24.2**	82.3	74.0	52.8	**74.8**	59.5	79.8	50.8	**66.7**
EvoLabel (original evolution)	71.6	80.3	59.8	84.7	18.9	83.8	76.8	47.0	72.2	51.3	75.3	21.2	61.9
EvoLabel Train-order ballot	**76.3**	79.8	61.2	84.0	22.8	**84.2**	77.5	48.7	72.5	54.8	75.3	42.8	65.0
BM25 third-arm vote	74.2	78.5	61.5	82.3	22.8	82.0	77.7	49.5	72.5	55.3	75.5	45.5	64.8
fixed: query-relevant arbiter	74.3	81.0	60.5	84.8	22.8	82.5	77.2	48.7	71.8	54.4	76.5	43.8	64.9
query-relevant arbiter + swap ballot	73.3	78.8	61.0	84.2	22.8	83.0	71.7	47.2	71.8	54.6	75.6	46.2	64.2
fixed: ensemble arbiter	74.2	**81.2**	61.2	84.7	23.2	83.2	74.4	48.9	72.4	55.2	76.1	43.8	64.9
dual-view + lexical voting (ensemble)	70.3	78.7	59.2	82.5	22.5	79.8	77.7	**53.7**	73.7	**61.0**	79.7	**51.5**	65.8
dual-view + lexical voting (3-way)	74.2	77.3	60.7	80.7	23.0	79.8	76.0	53.5	74.3	60.8	**83.7**	40.2	65.3
fixed: dual-view arbiter	73.2	80.0	**62.3**	82.8	22.8	83.0	78.0	51.3	74.8	59.3	81.2	49.8	66.5
MMR-diverse arms + ensemble arbiter	69.5	81.0	60.5	80.8	22.8	83.7	75.8	51.7	73.0	57.7	80.0	50.2	65.6
`scotta_style`	72.3	77.3	57.5	80.3	21.0	80.5	**78.2**	48.0	69.8	53.3	73.0	36.7	62.3
`static_topk`	72.3	74.7	61.7	79.2	23.0	80.3	76.3	48.3	71.8	53.7	74.3	43.7	63.3
`topk_sd`	63.3	78.5	59.2	76.0	22.7	74.5	72.0	48.5	71.0	56.2	74.7	40.8	61.4
`bm25`	66.2	67.8	57.7	74.8	22.2	74.2	70.2	50.2	66.8	57.8	71.0	39.1	59.8
frozen dual-arm (no evolution)	75.6	80.3	61.5	84.0	22.7	84.0	76.5	48.7	72.5	54.2	76.0	42.7	64.9
`random`	24.0	0.8	52.3	0.8	20.0	1.7	3.0	31.8	47.2	31.7	51.0	18.0	23.5

Table 13: **Multi-model robustness on `qwen2.5 32b-instruct` (17 methods × 12 datasets × 3 seeds).** Mean accuracy (%) over 3 seeds; best per column in bold, second underlined. Mirrors the main-study protocol (k=16, 500 train / 200 test). EvoLabel-Adaptive (ours) is the proposed adaptive router; its per-dataset breakdown is Table 14.

Method	agn	bank.	bq	clin.	csqa	hwu	liu	mnli	rte	snli	sst2	trec	mean
EvoLabel-Adaptive (ours)	90.5	88.2	74.8	**94.3**	84.5	91.2	**87.0**	86.3	**90.7**	**89.8**	95.0	79.2	**87.6**
EvoLabel (original evolution)	90.7	86.7	74.8	94.0	73.8	91.2	85.0	86.2	89.7	87.7	94.7	65.7	85.0
shipped single ballot	90.3	86.8	75.2	93.8	81.7	91.2	85.2	86.5	89.8	88.8	94.8	69.7	86.1
BM25 third-arm vote	90.3	85.5	75.0	92.2	83.5	91.0	85.8	85.5	89.8	89.2	94.7	72.0	86.2
fixed: query-relevant arbiter	90.7	88.2	75.3	94.0	80.7	91.3	86.7	85.7	90.0	89.2	94.7	72.2	86.6
query-relevant arbiter + swap ballot	90.2	88.2	75.2	94.2	81.8	91.2	86.7	86.3	90.0	89.2	94.7	71.2	86.6
fixed: ensemble arbiter	90.2	**88.3**	75.0	94.2	79.2	91.2	86.3	85.8	90.0	89.0	94.7	69.8	86.1
dual-view + lexical voting (ensemble)	90.0	86.5	74.3	93.0	84.5	**92.3**	86.7	**86.7**	90.2	89.3	**95.5**	**80.8**	87.5
dual-view + lexical voting (3-way)	90.0	83.7	74.7	91.7	85.3	90.8	84.7	**86.7**	90.2	89.8	94.8	75.0	86.5
fixed: dual-view arbiter	**91.2**	86.5	74.8	93.0	84.5	91.8	86.3	86.5	90.5	89.7	95.0	79.0	87.4
MMR-diverse arms + ensemble arbiter	90.2	87.3	**76.3**	93.3	84.0	92.2	85.3	86.5	90.0	89.5	**95.5**	74.5	87.0
scotta_style	90.5	86.2	72.5	91.7	**85.7**	90.3	84.8	85.5	89.8	87.8	94.5	74.0	86.1
static_topk	90.0	84.2	74.5	91.3	84.3	90.3	84.5	86.5	89.3	88.8	94.7	70.5	85.7
topk_sd	90.2	85.8	74.7	92.2	84.7	90.0	84.2	86.3	90.0	87.7	94.5	70.7	85.9
bm25	90.2	77.5	72.0	87.2	84.8	90.0	82.7	85.5	88.7	89.2	95.2	75.7	84.9
frozen dual-arm (no evolution)	90.5	86.7	74.8	94.0	82.5	91.2	85.5	86.2	89.5	89.0	94.7	70.5	86.3
random	24.0	0.8	52.3	0.8	20.0	1.7	3.0	31.8	47.2	31.7	51.0	18.0	23.5

The cut is unidentifiable, not validated. Pool inventories contain 2–6 labels for the eight non-intent tasks and 50–147 for the four intent tasks (nominal 2–6 versus 54–150; Table 8), and no dataset lies between them. Every threshold in $[7, 50]$ therefore induces the identical partition, and this cuts both ways. On the one hand the value 20 cannot have been fitted to accuracy, since varying it changes no reported decision and a sweep inside the interval would reproduce the same table by construction. On the other hand these data cannot *test* the cut: validating it would require tasks with intermediate cardinalities, on which it becomes a genuine hyperparameter demanding held-out dataset-level calibration. We therefore describe the router as *two expert configurations indexed by task family*, chosen on this suite, rather than as a validated policy; this is one reason why Section 7.2 does not present cross-model results as unseen-dataset validation.

The gate excluded nothing here. The gate of Eq. 3 is evaluated once per dataset from pool labels and evaluated open on all twelve datasets, so no dataset was excluded from any comparison and the 0.70 level did not select among the reported systems. We do not report per-dataset margins above the level, so this is a statement that nothing was excluded rather than a demonstration of a comfortable margin on every task; on a sparser pool the level would bind.

Table 14: **Per-dataset breakdown of EvoLabel-Adaptive on the three additional open-weight models.** Top panel: accuracy (%) of EvoLabel-Adaptive; bottom panel: accuracy difference versus the frozen dual-arm `frozen_dualarm`, with * marking a significant continuity-corrected paired McNemar test (p<0.05, per-query correctness joined on query text, n=600 per dataset per model). Datasets are grouped into the large-$|\mathcal{Y}|$ intent sets, where the adaptive router dispatches to the online-evolution expert (Ballot-ensemble arbiter), and the small-$|\mathcal{Y}|$ sets, where it dispatches to the orthogonal-lexical challenger (Orthogonal-lexical challenger). Across the 36 dataset×model cells the adaptive method is significantly *better* than the frozen dual-arm in 12 cells, significantly *worse* in 1 (the fixed arm remains slightly stronger on Mistral/LIU54), and at parity in the remaining 23; the pooled advantage is nevertheless significant on all three models (bottom row). The mechanism shows up in the group means (large-$|\mathcal{Y}|$ vs small-$|\mathcal{Y}|$ deltas): llama3.1 +2.71 vs +0.96; mistral −0.71 vs +2.98; qwen2.5 +0.83 vs +1.65. On Mistral and Qwen2.5 the adaptive router mainly recovers small-$|\mathcal{Y}|$ ground the fixed Query-relevant ballot arbiter had left on the table, while on Llama the large-$|\mathcal{Y}|$ Banking77 and CLINC150 sets contribute as well.

Dataset	$\lvert\mathcal{Y}\rvert$	Accuracy (%)		
		`llama3.1:8b`	`mistral:7b`	`qwen2.5:32b`
Large $\lvert\mathcal{Y}\rvert$ (intent), routed to the online-evolution expert				
Banking77	77	77.50	80.83	88.17
CLINC150	150	85.83	84.83	94.33
HWU64	64	85.17	82.33	91.17
LIU54	54	80.17	74.00	87.00
Small $\lvert\mathcal{Y}\rvert$, routed to the orthogonal-lexical challenger				
AG-News	4	76.33	74.33	90.50
TREC	6	68.67	50.83	79.17
CommonsenseQA	5	22.83	24.17	84.50
MNLI	3	45.50	52.83	86.33
SNLI	3	43.33	59.50	89.83
RTE	2	62.17	74.83	90.67
BoolQ	2	56.00	62.17	74.83
SST-2	2	83.50	79.83	95.00
All 12 (pooled)	–	65.58	66.71	87.62
Δ vs `frozen_dualarm` (pp)				
Large $\lvert\mathcal{Y}\rvert$ (intent)				
Banking77		+6.00*	+0.50	+1.50*
CLINC150		+2.33*	+1.17	+0.33
HWU64		+1.67	−2.00	+0.00
LIU54		+0.83	−2.50*	+1.50
Small $\lvert\mathcal{Y}\rvert$				
AG-News		+0.17	−1.17	+0.00
TREC		+3.00*	+8.17*	+8.67*
CommonsenseQA		+0.50	+0.83	+2.00
MNLI		−0.33	+4.17*	+0.17
SNLI		+0.67	+5.33*	+0.83
RTE		+1.67	+2.33*	+1.17*
BoolQ		−0.67	+0.33	+0.00
SST-2		+2.67*	+3.83*	+0.33
All 12 (pooled)		+1.54*	+1.75*	+1.38*

The assignment is supported on both branches. Holding the dual-arm machinery frozen and varying only the arbiter, the ballot-ensemble arbiter deployed on the many-class branch improves over the shipped train-order alternative by +0.55 pp (paired McNemar $p = 0.0015$, positive on 14/16 cells; Table 18). Both fixed experts were also run on all twelve tasks, so each branch can be checked directly against its alternative. On the four many-class tasks the ballot-ensemble arbiter scores 85.96% against 85.50% for the lexical-challenger configuration; on the eight low-cardinality tasks the ordering reverses, 78.67% versus 75.88%. The adaptive configuration matches the better

Table 15: **Reported paired McNemar comparisons.** Adaptive versus four comparators, joined by query text across 12 datasets and three seeds. Nominal pooled counts are 21,600 predictions (7,200 per model); repeated-query and task dependence are not cluster-adjusted in these reported tests. Differences use unrounded execution-level means.

	vs `static_topk`	vs `scotta_style`	vs `frozen_dualarm`	vs fixed: query-relevant arbiter
`llama3.1:8b`	+3.39 pp, $p<0.001^*$	+2.96 pp, $p<0.001^*$	+1.54 pp, $p<0.001^*$	+0.75 pp, $p=0.007^*$
`mistral:7b-instruct`	+3.49 pp, $p<0.001^*$	+4.38 pp, $p<0.001^*$	+1.75 pp, $p<0.001^*$	+1.60 pp, $p<0.001^*$
`qwen2.5:32b-instruct`	+1.88 pp, $p<0.001^*$	+1.51 pp, $p<0.001^*$	+1.38 pp, $p<0.001^*$	+1.08 pp, $p<0.001^*$
Pooled	+2.92 pp, $p<0.001^*$	+2.95 pp, $p<0.001^*$	+1.56 pp, $p<0.001^*$	+1.14 pp, $p<0.001^*$
$\lvert\mathcal{Y}\rvert \geq 20$ (intent)	+4.75 pp, $p<0.001^*$	+3.74 pp, $p<0.001^*$	+0.94 pp, $p<0.001^*$	+0.11 pp, $p=0.653$
$\lvert\mathcal{Y}\rvert < 20$ (non-intent)	+2.00 pp, $p<0.001^*$	+2.56 pp, $p<0.001^*$	+1.86 pp, $p<0.001^*$	+1.66 pp, $p<0.001^*$

Table 16: **Earlier four-task study.** Accuracy and macro-F_1 over two seeds with 200 queries per cell on Qwen3-8B. These historical runs are distinct from the three-seed main grid; values are retained from the original experimental summary.

Method	bank.	clin.	hwu64	liu54	Acc μ / mF$_1$ μ
static_topk (single-arm)	80.0	85.0	83.3	83.3	82.9 / 79.4
SCOTTA-style (selection-side)	79.8	85.0	83.3	83.3	82.8 / 79.4
EvoLabel (dual-arm, frozen)	82.5	89.3	85.0	84.7	**85.4 / 83.0**
EvoLabel (full, + evolution loop)	82.5	89.0	85.0	84.7	85.3 / 83.0

expert on each side to within 0.05 pp (86.00% and 78.71%), which also verifies that the routing is implemented as specified. Substituting the wrong expert would cost 0.46 pp on the many-class tasks and 2.83 pp on the low-cardinality ones, the latter consistently across all eight tasks (8/8, Wilcoxon $p = 0.0078$). Both branches are therefore supported by a direct comparison rather than chosen arbitrarily, although the assignment is still selected on this suite. The ballot count remains a fixed budget rather than a swept parameter: it enters cost linearly through Eq. 7, so b_A is a cost–accuracy knob deliberately held at one value per branch.

What the reflector constants do and do not establish. Zero operations were committed in all audited records (36 six-ballot, 36 train-order and 36 adaptive), so no reflector threshold can have produced an accuracy difference *among the systems compared here*: the branch they govern never executed. That is a claim about internal validity, and it does not settle a stronger question. Whether a more permissive schedule—lower support, a larger budget, a longer window, or a different proposal distribution—would activate operations from label-free feedback alone is untested, because these thresholds were held fixed. The reported null is therefore scoped to the configured mechanism at these settings, exactly as Section 5 frames it conditionally, and is not evidence about label-space evolution in general. Section A.18 records the same caution for the supervised direction: an earlier gold-probe improvement claim was not reconciled with the variant table and is not retained. A threshold sweep with matched execution is the natural next experiment.

Shared protocol constants. $k = 16$, the 500-example pool and the 200-query test sample are identical for all 24 methods, as are the prompt template, the decoder and the canonical label order. The last of these is itself a controlled choice rather than an arbitrary default: presentation order alone shifts accuracy by up to 7.5 pp on Banking77, so fixing one canonical sorted order for every method removes a confound that would otherwise be larger than the effects under study (Appendix A.6). Because no method receives a different budget, these values cannot advantage the proposed configuration; they fix the comparison protocol instead of tuning a system.

A.12 Routing and reproducibility limitations

The cardinality threshold and expert pair were calibrated on this benchmark suite; Appendix A.11 records every constant and shows why the cut is invariant within the 6–54 nominal-label gap. A threshold sweep inside that gap would therefore leave every routing decision unchanged and would not independently validate routing. Model transfer reuses the same tasks. Dataset-level holdout or

Method	Banking	Clinc	HWU64	Liu54	Mean
Frozen dual-arm (the honest null)	82.25	89.50	85.25	84.75	85.4
EvoLabel train-order ballot (label-free)	85.25	88.25	86.00	81.25	85.19
EvoLabel train-order ballot + Merge	85.75	88.25	86.25	81.00	85.31
EvoLabel train-order ballot + Describe	85.25	88.25	86.25	80.75	85.12
EvoLabel train-order ballot, gold-10 probe	85.25	87.50	86.00	80.25	84.75
EvoLabel train-order ballot, gold-100 probe	85.25	88.25	85.75	81.25	85.12
EvoLabel train-order ballot, gold-10 + warmup	85.25	88.25	86.00	81.25	85.25
EvoLabel train-order ballot, gold-100 + warmup	85.50	88.50	86.25	80.50	85.06
EvoLabel BM25 third-arm vote (lexical 3rd arm)	85.00	88.50	84.25	81.00	84.62
EvoLabel BM25 third-arm vote, gold-100	84.50	88.25	84.25	80.50	84.44
EvoLabel BM25 third-arm vote, same-LLM runoff	85.25	88.50	85.25	80.50	84.88
EvoLabel query-relevant ballot	86.50	89.50	81.75	82.25	85.00
EvoLabel query-relevant ballot, order swapped	86.25	89.00	81.00	82.25	84.62

Table 17: **Earlier evolution variants.** Reported accuracy (%) on four intent datasets. These configurations include different settlement rules, gold-probe rates and rewrite settings; comparisons are system-level unless otherwise controlled. The frozen dual-arm row is the reference for assessing rewrite effects. These historical values are distinct from the three-seed main grid.

Table 18: Arbiter refinement at the extended protocol (500 test queries × four seeds = 8,000 predictions; `qwen3:8b`; the four intent datasets). The dual-arm method is held frozen; only the runoff arbiter changes. The ballot-ensemble arbiter (six query-relevance slices, stronger arm presented first, majority vote) adds +0.55 pp over the shipped train-order runoff (paired McNemar p=0.0015, highly significant) and is positive on 14/16 cells. This gain is an *arbiter* effect inside the dual-arm machinery, orthogonal to the (null) evolution layer (Tab. 17).

Arbiter	banking	clinc	hwu64	liu54	mean
Train-order ballot (shipped)	83.8	88.6	87.7	81.9	85.5
Ballot-ensemble arbiter	83.9	89.3	88.2	82.9	86.0
Δ (ballot-ensemble − train-order)	+0.1	+0.7	+0.5	+0.9	+0.55

nested dataset-level selection would address a different, stronger generalization question, which these results do not supply.

The supplementary multi-model provenance ledger records adopted execution cells. Served-model re-sampling can change individual outputs, so matching configuration identifiers does not guarantee bitwise replication. The manuscript reports only anonymous, relative provenance. The supplementary artifact replaces private node names and source paths with anonymous execution identifiers.

A.13 Extended related-work positioning

The object of adaptation. Example selection, label presentation and accumulated memory are different interventions. Similarity retrieval chooses relevant examples (Liu et al., 2022), diversity-based selection broadens coverage (Levy et al., 2023), and selective annotation changes the labeled pool (Su et al., 2023). Instance-specific ordering and gradient-based selection change how the pool is used (Wu et al., 2023; Zhang et al., 2025b). We keep the evidence pool fixed. Results under this restriction should not be read as a ranking of every online adaptation method or its original implementation.

Presentation is not an information-theoretic equivalence. Label tokens (Wang et al., 2023a), verbalizer knowledge (Hu et al., 2022), candidate ordering (Zheng et al., 2024), and prompt permutations (Lu et al., 2022) can affect model behavior. Even when two representations carry equivalent information for an ideal observer, a fixed finite model need not process them equally well. This is why the gain decomposition explicitly assumes outside-subset invariance rather than deriving it from the fact that rewrites use an existing pool. Contextual calibration (Zhao et al., 2021) addresses answer bias through another intervention and requires a separately controlled comparison.

Table 19: **Earlier arbiter summary.** Reported accuracy and macro-F_1 for three settlement configurations. This archived summary is distinct from the 36 six-ballot runs used in the trace-derived figures; the original claim of identical inputs across all configurations has not been re-established here.

	CORE (4)		REST (8)		All (12)	
Variant	Acc	mF_1	Acc	mF_1	Acc	mF_1
Train-order ballot (shipped)	85.0	81.4	77.1	76.6	79.7	78.2
Query-relevant ballot	85.3	81.7	76.8	76.7	79.7	78.4
Ballot-ensemble arbiter (deployed)	86.0	82.4	77.0	76.8	80.0	78.7

Table 20: Confusion audit (Section 7.4). Top-5 confusable label pairs are *discovered* on the seed-42 streams and evaluated on the disjoint seed-43 streams (held-out queries, n_{held}). "top-K err" is the model's error rate on queries whose gold class is one of the pair; "matched" is the same quantity for frequency-matched random class pairs (200 permutations); "corpus avg" is the marginal error rate. p is the one-sided permutation p-value for *matched* $\leq$ *top-K*; no dataset is significant. "oracle P@5" is the precision of the discovered pairs at predicting the true confusion pairs of the held-out stream. No chance-level calibration is assumed.

Dataset	n_{held}	top-K err%	matched%	corpus%	p	oracle P@5
banking	190	30.00	33.36	17.36	0.677	0.00
clinc	108	28.70	37.90	14.50	0.756	0.00
hwu64	238	38.66	36.73	19.36	0.383	0.20
liu54	637	29.51	36.26	19.79	0.632	0.20
POOLED	1173	31.37	36.06	17.75	—	0.10

Accumulated memory and a fixed inventory. ACE and Dynamic Cheatsheet retain and update contextual content (Zhang et al., 2026; Suzgun et al., 2026). Our canonical output classes and pairwise update mechanism are constrained. Failure to accumulate enough pair evidence here does not contradict successful memory adaptation elsewhere. The important distinctions are the state being updated, available feedback, and whether the evaluated implementation actually changes that state.

Arbitration versus abstention. Self-consistency aggregates alternative sampled answers (Wang et al., 2023b); Bayesian prompt ensembles model uncertainty across prompts (Tonolini et al., 2024). Neither is identical to choosing between two retrieval-conditioned predictions. Selective classifiers optimize or calibrate rejection (Geifman & El-Yaniv, 2017; 2019), while trust scores and language-model calibration use other signals (Jiang et al., 2018; 2021). Our two operating points show the empirical effect of rejecting disagreements; they do not establish calibrated probabilities, a continuous risk–coverage curve, or dominance over these methods.

Selection data and evaluation data. Inference-time label access and development-time model selection are distinct (Perez et al., 2021). Our threshold and expert pair were chosen on the suite, although the deployed rule reads only cardinality. Additional backbones test model transfer with tasks fixed. Unseen-task transfer would require selecting the configuration without using the held-out task, rather than repeating the same configuration with new model outputs.

A.14 Prediction procedure and state boundaries

Algorithm 1 separates the training-pool gate, per-query agreement test and optional round-level reflector. The gate uses leave-one-out retrieval coverage on the labeled pool. Agreement is assessed after resolving presentation entries to canonical output classes. Reflection proposes changes for subsequent queries and does not retroactively alter a scored prediction.

Canonical versus presented labels. A merged entry can refer to more than one original class; accuracy is still scored in the original task space. String equality is therefore not a general definition

Table 21: Selective prediction on the disagreement signal (Section 7.3). Per variant, pooled over four datasets and two seeds (n=1600). "Full" is accuracy on all queries; "cov." is the fraction on which the arms *agree* and the system answers; "agreed" / "disagreed" are accuracies on the two subsets; gain = selective minus full accuracy, obtained with *zero* extra LLM calls.

Variant	Full%	cov.%	agreed%	disagreed%	gain
Train-order ballot (label-free)	85.19	86.31	91.24	47.03	+6.05
Label-rewrite, gold-10	84.75	86.50	90.61	47.23	+5.86
Label-rewrite, gold-100	85.12	86.50	91.11	46.76	+5.99
Label-rewrite, gold-100 + warmup	85.06	86.12	91.22	46.85	+6.16
BM25 third-arm vote (BM25 vote3)	84.62	86.88	90.86	43.33	+6.24
BM25 third-arm, gold-100	84.44	86.38	91.17	41.74	+6.73
BM25 third-arm, same-LLM runoff	84.88	87.25	90.54	46.07	+5.67
POOLED (n=11200)	84.87	86.56	90.96	45.58	+6.10

Table 22: **Earlier matched-coverage comparison.** Reported selective accuracy of dual-arm agreement and a retrieval-margin rule. The execution snapshot differs from the six-ballot runs in Figures 1 and 3; these values must not be pooled across snapshots.

Dataset	Cov. (%)	Full (%)	Dual-arm agree (%)	Retrieval-margin baseline (%)	Oracle (%)
banking	85.2	84.3	88.2	84.9	99.0
clinc	87.3	88.8	94.5	89.7	99.4
hwu64	87.2	87.0	91.9	88.7	99.6
liu54	87.3	83.2	88.0	84.0	95.4
ag_news	93.2	86.3	87.8	86.0	92.7
boolq	99.5	64.2	64.3	64.0	64.5
commonsenseqa	86.3	60.0	65.8	60.6	69.5
mnli	96.8	74.8	74.7	74.5	77.3
rte	99.5	82.8	82.9	82.9	83.3
snli	97.5	78.8	78.8	79.4	80.9
sst2	99.8	94.5	94.5	94.5	94.7
trec	85.8	71.0	75.9	72.0	82.7

of canonical agreement. The new selective figures use the logged `resolved_disagreed` flag and check query-trace alignment with scored rows. No edits occur in those 36 runs, simplifying the state trajectory but not removing the need for consistent definitions.

Inference costs. A closed gate requires one F prediction; an open gate evaluates two views. Disagreement can add up to six calls for the ensemble and twelve for the dual-view arbiter. These are additional calls, not a two-call end-to-end budget. Logged runtime includes serving and implementation effects, whereas call counts describe the algorithmic path. Reflector overhead must also be included when an evolution-enabled configuration actually invokes reflection. The runtime scatter is descriptive rather than a controlled hardware-normalized ranking.

A.15 Variant definitions and ablation scope

Frozen dual-arm. Keeps presentation state fixed and provides the direct reference for testing whether executed edits improve predictions.

Train-order ballot. Builds arbitration slates from candidate-class examples in training order. This comparator is included in the restored 24-method main table.

Query-relevant ballot. Uses query similarity to select candidate-class examples; an order-swapped variant probes presentation sensitivity.

Ballot ensemble. Combines six query-relevance ballots. Standalone executions and executions inside Adaptive are separate runs: nominally matching configuration does not guarantee identical sampled outputs.

Table 23: **Design constants.** *Shared* means the value is identical for all 24 compared methods; *non-binding* means no reported comparison changes when the value varies over its stated range on this suite.

Constant	Value	Status / justification
Cardinality cut	$\lvert\mathcal{Y}\rvert \geq 20$	invariant: any value in $[7, 50]$ gives identical routing
Coverage gate	$C(P) \geq 0.70$	non-binding: open on all twelve datasets
Reflector support	≥ 4	non-binding: zero operations committed
Operation budget	≤ 8	non-binding: zero operations committed
Reflector window	≤ 64	non-binding: zero operations committed
Ballot budget	$b_A \in \{6, 12\}$	fixed a priori; bounds disagreement cost via Eq. 7
k, pool, test size	16, 500, 200	shared by every method; cannot favour one method

Algorithm 1 Prediction with a separately evaluated update layer

Require: Query x, fixed pool P, canonical inventory $\mathcal{Y}$, state L_t
Select second view S and arbiter A using the fixed cardinality rule
Predict $\hat{y}_F$ using semantic top-k retrieval and static L_0
if training-pool coverage is below 0.70 **then**
  return $\hat{y}_F$
end if
Predict $\hat{y}_S$ using the selected second view
Resolve both predictions to canonical classes
if the predictions agree **then**
  $\hat{y} \leftarrow \hat{y}_F$
else
  $\hat{y} \leftarrow A(x, P, \hat{y}_F, \hat{y}_S)$
end if
Finalize and score $\hat{y}$; append $(\hat{y}_F, \hat{y}_S)$ to the feedback window
if evolution is enabled and the update boundary is reached **then**
  Rank an operation o_t from pair support and label similarity
  if o_t passes support, type, and state-consistency checks **then**
    $L_{t+1} \leftarrow \Phi(L_t, o_t)$
  else
    $L_{t+1} \leftarrow L_t$
  end if
end if
return $\hat{y}$

Lexical challenger. Uses a BM25 second view and a dual-view arbiter, changing retrieval as well as disagreement resolution.

Gold probes. Supply labels to the reflector for diagnostic purposes; these supervised variants cannot establish label-free deployment performance.

Merge and Describe. Alter grouping or descriptive content. An operator that never commits an update cannot explain an observed prediction gain.

A single-module comparison requires matched query streams, base predictions, retrieved evidence, candidate sets and decoding, apart from the named change. End-to-end runs can differ on several of these dimensions. The earlier extended arbiter study (Table 18) uses 500 queries and four seeds; its reported effect concerns arbitration, not executed label edits. The archived three-variant summary (Table 19) does not reproduce the means of the current standalone six-ballot pool. These snapshots are not combined into one monotonicity or significance claim.

A.16 Metric definitions and aggregation

For n queries with correctness c_i and operational disagreement d_i,

$$a = \frac{1}{n}\sum_i c_i, \qquad \delta = \frac{1}{n}\sum_i d_i, \qquad a_{\text{agree}} = \frac{\sum_i (1-d_i)c_i}{\sum_i (1-d_i)}.$$

Agreement-only coverage is $1-\delta$; selective gain is $a_{\text{agree}} - a$. A conditional accuracy with zero denominator is undefined, not zero; every plotted subset has a nonzero denominator. Macro-F_1 in archived tables is a separate class-averaged metric and cannot be reconstructed from accuracy alone.

Seed-first summaries. For accuracy a_{mds} of method m, dataset d and seed s, the main table uses $\bar{a}_{md} = \frac{1}{3}\sum_s a_{mds}$, then equally averages the 12 datasets. Cell variability is sample SD, $\sqrt{\sum_s (a_{mds} - \bar{a}_{md})^2/2}$. For method ranking, compute $b_{ms} = \frac{1}{12}\sum_d a_{mds}$ and the SD of its three seed values. The mean of cell SDs is not the SD of an overall mean. Gain error bars are computed from paired differences $a_{mds} - a_{bds}$ rather than by adding marginal error bars. These descriptive SDs do not quantify unseen-task transfer.

Oracle accuracy versus recovered gain. An oracle that selects a correct arm whenever possible bounds only rules restricted to those arm outputs. The ratio $a_{\text{arbiter}}/a_{\text{oracle}}$ is relative accuracy, not the fraction of gain recovered over F. The latter is $(a_{\text{arbiter}} - a_F)/(a_{\text{oracle}} - a_F)$ when its denominator is positive. Earlier recovery graphics conflated these interpretations and are not used to support the revised headline claims.

A.17 Operator-audit controls

The archived audit selects top-five pairs on discovery streams. Evaluation compares errors on queries associated with those pairs to corpus error and to frequency-matched random pairs, with 200 permutations. A large marginal contrast can reflect difficult or frequent classes; matched controls address part of this selection effect. The reported permutation values, 0.383–0.756, do not support excess error relative to those controls. Failure to reject is not proof that all operators are ineffective.

Different seeds alone do not establish stream disjointness. The original audit is reported as using disjoint queries; reproducing that claim requires actual query identities, overlap checks and discovery/evaluation manifests. These are separate from the newly checked 864-cell manifest. Neither pair transfer nor a zero-operation replication validates the counterfactual effect of an unexecuted rewrite.

A.18 Taxonomy auditing and supervised extensions

Logs could support human taxonomy review by retaining triggering queries, canonical predictions, retrieved evidence, proposed operations and rejection reasons. A reviewer can distinguish semantic overlap from retrieval failure or an underspecified description. This is a possible diagnostic application, not an empirically validated autonomous editing system. Plausible pair names alone do not justify merging canonical task classes.

A supervised probe introduces additional information and tests whether more reliable feedback improves update selection. Such an experiment should fix the prediction stream, disclose its label budget and include a supervised non-rewriting comparator. An earlier gold-probe improvement claim was not reconciled with the variant table and is not retained as an established result. The tabulated gold-probe variants remain explicitly supervised. Independent taxonomies, changing inventories and new evidence are useful future settings, provided both accepted and rejected operations are logged.

A.19 Reproduction workflow and source hierarchy

The main table, seed-SD table and result panels are generated by `code/draw_results.py`; the architecture and motivation use `draw_architecture.py` and `draw_motivation.py`. The result renderer checks one record per method–dataset–seed cell, verifies 200 scored rows and recomputes accuracy from correctness flags. Its manifest retains relative paths and SHA256 digests for all 864 adopted inputs. Cross-model plotted differences use the adopted-accuracy ledger rather than rounded

table entries. Earlier diagnostic tables remain archived summaries, with their provenance limitations stated separately.

The evidence hierarchy is explicit: checked raw records support main-grid means and figures; the adopted ledger supports cross-model summaries; historical tables support their identified protocols. Reconstructing an old snapshot requires code and query identities, not just matching method names. Re-sampling a served model can change predictions at the same nominal settings. A release should retain input hashes, run identifiers, invalid-output rates, model versions, decoding and environment information, while removing private machine names and user directories from public provenance.