ADALABEL: LEARNING SIGNED ANNOTATION VALUE FOR BUDGETED DATA ANNOTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Stronger annotators can improve labels on average without being correct on every example. Replacing a cheap label can rescue an error or regress a correct annotation. We therefore define *signed annotation value* (SAV) as the expected change in annotation correctness caused by an upgrade, conditional only on information available before the strong call. AdaLabel learns the CC/WC/WW/CW outcome distribution from paired Meta supervision and ranks upgrades by $V(x) = P(WC \mid s_{cheap}(x)) - P(CW \mid s_{cheap}(x))$. In an independently and prospectively frozen Yahoo Answers evaluation, AdaLabel improves over the prespecified Direct MLP signed-value comparator by 73–170 net corrections under matched logical-token budgets, while remaining competitive with strong heuristic alternatives. An exploratory frozen-record replay recovers WC-only under constant predicted regression risk and produces dataset-dependent ranking divergence as heterogeneity is restored, without guaranteeing utility gains. Exploratory frozen-record attribution supports instance-level value learning beyond class-only priors, and independent Amazon results show that retaining four outcomes improves joint-failure auditability. Results vary across budgets and datasets, so they do not establish universal dominance.

1 INTRODUCTION

Cheap LLM annotation can cover a large training population, while stronger annotation requires more compute. Under a limited budget, the practical decision is which existing cheap labels to replace before training a downstream model. Active learning motivates allocating scarce labeling calls selectively (Lewis & Gale, 1994); here, however, every example already has an annotation. The decision concerns the value of changing its source.

Stronger on average does not mean beneficial on every example. In DBPedia14, cheap and strong accuracy are 50.15% and 62.82%, yet the same population contains 870 rescues and 361 regressions. We denote these outcomes WC (cheap wrong, strong correct) and CW (cheap correct, strong wrong). Uncertainty measures ambiguity, cheap-error likelihood includes errors that the stronger annotator also makes, and disagreement groups rescues with regressions. Each signal can inform allocation, but none by itself measures the value of actually replacing a label with a non-oracle annotation.

We therefore formulate annotation upgrading as a signed decision. The correctness change ΔQ is +1 for WC, -1 for CW, and 0 for joint correctness (CC) or joint failure (WW). Its prospective signed annotation value, conditioned on information available before the strong call, is

$$V(x) = \mathbb{E}[\Delta Q \mid s_{cheap}(x)] = P(WC \mid s_{cheap}(x)) - P(CW \mid s_{cheap}(x)).$$

This quantity asks what correctness change is expected if the upgrade is performed: expected rescue minus expected regression. It distinguishes examples that are equally uncertain or equally likely to have a wrong cheap label but differ in their likely response to replacement. The scientific object is the annotation-source intervention during training-data construction. This differs from inference-time selection or conditional computation (Geifman & El-Yaniv, 2017; Graves, 2016) and from valuing an example's contribution to learning (Ghorbani & Zou, 2019).

AdaLabel provides a concrete realization of this formulation (Figure 1). Paired cheap and strong annotations with reference labels on a Meta set supervise the four-state outcome distribution. At

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.

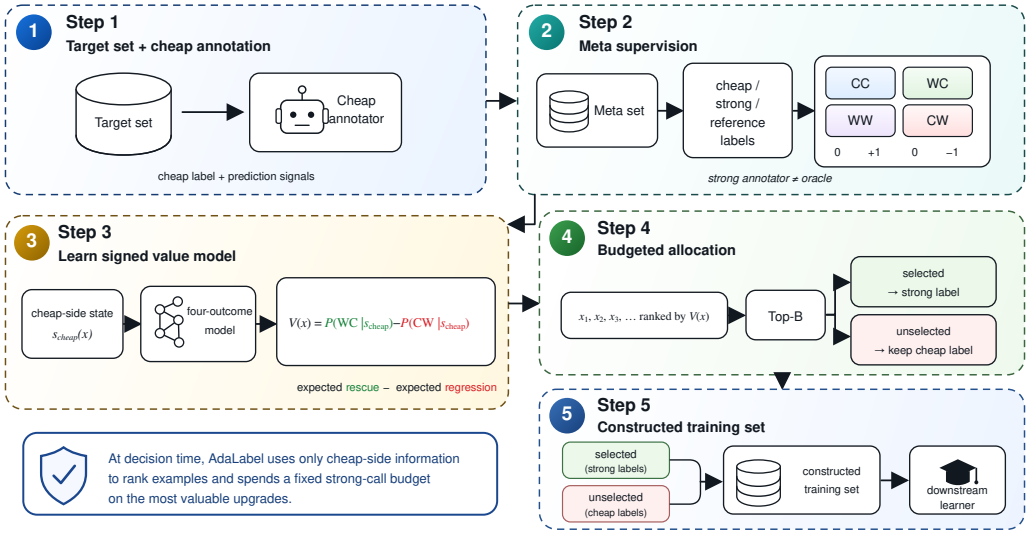

Figure 1: AdaLabel allocation framework. Paired Meta supervision learns a signed annotation-value model from cheap, strong, and reference annotations. On the target population, ranking uses cheap-side state only: the Top-B examples receive strong annotations and all others retain cheap labels.

allocation time, the model uses cheap-side state to estimate these probabilities, derives their signed contrast, and selects the Top-B examples. AdaLabel is estimator-agnostic. A lightweight multinomial model isolates the SAV formulation from model capacity. Retaining the four outcomes also makes joint success and joint failure available for inspection even though both have zero signed utility.

The experiments combine prospective Yahoo Answers confirmation under matched logical-token budgets with frozen-record attribution, an independent Amazon auditability test, and supporting comparator, downstream, and transfer evidence. Together, they test when signed allocation is useful and where its effects remain heterogeneous.

Our contributions are:

- **Formulation.** We define signed annotation value for budgeted label upgrading with a stronger but non-oracle annotator, accounting jointly for rescue and regression; we characterize its equivalence to WC-only under constant regression risk and divergence under heterogeneous, predictable risk.
- **Method.** AdaLabel is one estimator of this decision quantity: it estimates the CC/WC/WW/CW distribution from paired Meta supervision and ranks upgrades by $P(WC \mid s_{cheap}) - P(CW \mid s_{cheap})$.
- **Evidence.** An independently prospective, resource-matched Yahoo Answers evaluation is complemented by attribution, auditability, downstream, and transfer analyses that delineate the method's scope.

2 Related Work

2.1 Annotation Quality, Weak Supervision, and LLM Annotators

Crowdsourcing and multi-annotator learning show that annotation sources have heterogeneous, instance-dependent error patterns and that repeated labeling is not uniformly beneficial (Dawid & Skene, 1979; Snow et al., 2008; Sheng et al., 2008; Whitehill et al., 2009; Raykar et al., 2010; Rodrigues & Pereira, 2018). Weak-supervision systems instead combine noisy labeling functions into probabilistic training labels (Ratner et al., 2016; 2020), while benchmark audits demonstrate why label quality itself deserves explicit treatment (Northcutt et al., 2021). These approaches estimate latent labels or source reliability. AdaLabel assumes a fixed cheap/strong hierarchy and asks a

different prospective question: for which already cheaply labeled examples is replacement expected to improve correctness?

LLMs now generate and refine training annotations, with substantial variation across tasks (Zhang et al., 2023; Lee et al., 2023; Ding et al., 2023; He et al., 2024; Tan et al., 2024; Gilardi et al., 2023). CoAnnotating allocates work using uncertainty (Li et al., 2023), while LMCrowd selects relabeling examples within an LLM hierarchy (Zhang et al., 2025). AdaLabel instead learns signed upgrade value, treating rescues positively and regressions negatively; higher aggregate strong-annotator accuracy therefore need not make every upgrade beneficial.

2.2 Active learning and budgeted subset selection

Active learning selects examples for new labels using information gain, committee disagreement, uncertainty, or expected error reduction (MacKay, 1992; Seung et al., 1992; Lewis & Gale, 1994; Cohn et al., 1994; 1996; McCallum & Nigam, 1998; Tong & Koller, 2001; Roy & McCallum, 2001). Deep uncertainty acquisition and coreset methods extend this budgeted selection principle (Gal et al., 2017; Sener & Savarese, 2018). AdaLabel shares the pool-and-budget structure and uses uncertainty-related cheap-side features, but the action differs: every example already has a label, and querying replaces it with another non-oracle label. The target is therefore neither informativeness nor representativeness, but $\mathbb{E}[\Delta Q \mid s_{cheap}]$, which explicitly separates rescue from regression. Learning that relation requires paired Meta outcomes and introduces a transfer assumption absent from standard oracle-query formulations.

2.3 Selective prediction, learning to defer, and triage

Reject-option and selective-classification methods trade coverage for predictive risk (Chow, 1970; Bartlett & Wegkamp, 2008; El-Yaniv & Wiener, 2010; Wiener & El-Yaniv, 2011; Geifman & El-Yaniv, 2017; 2019; Cortes et al., 2016). Learning-to-defer extends abstention by jointly deciding whether a model or external expert should predict, with work on consistent losses, calibration, multiple experts, and exact assignment (Madras et al., 2018; Mozannar & Sontag, 2020; Verma & Nalisnick, 2022; Mozannar et al., 2023; Verma et al., 2023). Algorithmic triage and human-assistance systems likewise optimize instance-level division of labor (Okati et al., 2021; De et al., 2021), while complementary-team objectives show that the most accurate component need not maximize system utility (Wilder et al., 2020; Bansal et al., 2021).

This literature is the nearest conceptual neighbor, but ordinarily assigns responsibility at prediction time. AdaLabel instead acts during training-data construction: selected examples receive strong labels, the remainder retain cheap labels, and utility is evaluated at matched annotation budgets.

2.4 Conditional computation and data valuation

Cascades and conditional-computation methods allocate inference compute by input (Viola & Jones, 2001; Teerapittayanon et al., 2016; Graves, 2016; Bolukbasi et al., 2017; Shazeer et al., 2017; Kaya et al., 2019); data valuation measures an example's contribution to a fitted model (Koh & Liang, 2017; Ghorbani & Zou, 2019). AdaLabel instead values an annotation-source intervention, with direct correctness change measured by $U(S) = \#WC(S) - \#CW(S)$ and downstream performance evaluated separately.

3 Method

3.1 Budgeted annotation upgrading

The target allocation population is $D_T = \{(x_i, a_c(x_i))\}_{i=1}^N$: at allocation time it contains an input and its cheap annotation, together with cheap-side prediction state. A strong annotator can produce $a_s(x)$ as an upgrade, but is not an oracle. An allocation policy $\pi(x) \in \{0, 1\}$ chooses whether an example receives that upgrade under budget B:

$$\sum_{i=1}^{N} \pi(x_i) \leq B.$$

All policies receive the same B strong calls; this controls call count, not wall-clock time, price, token use, or latency.

3.2 SIGNED UPGRADE OUTCOMES

For a paired Meta or evaluation example with reference label y, comparing the cheap and strong annotations against y yields four mutually exclusive outcomes: CC (both correct), WC (cheap wrong, strong correct), WW (both wrong), and CW (cheap correct, strong wrong). WC is a rescue and CW is a regression. We encode the correctness change as

$$\Delta Q(x) = \begin{cases} +1, & WC, \\ -1, & CW, \\ 0, & CC \text{ or } WW. \end{cases}$$

3.3 SIGNED ANNOTATION VALUE

Let $s_{cheap}(x)$ denote the information available before a strong call. Signed annotation value (SAV) is

$$\begin{aligned} V(x) &= \mathbb{E}[\Delta Q(x) \mid s_{cheap}(x)] \\ &= P(WC \mid s_{cheap}(x)) - P(CW \mid s_{cheap}(x)). \end{aligned}$$

The CW term follows from signed utility rather than a post-hoc penalty. Unlike uncertainty, $V(x)$ estimates the correctness change produced by replacement and can prefer a moderately uncertain example with a better rescue-to-regression balance.

Cheap-error probability groups WC rescues with WW joint failures, while disagreement groups beneficial WC with harmful CW. The four-class representation keeps these cases distinct and derives a scalar score only after estimating their conditional probabilities.

3.4 LEARNING FROM PAIRED META OUTCOMES

Cheap-side state. The allocation state contains only pre-upgrade cheap-annotator and task information: uncertainty, confidence and concentration, score dispersion and energy, a Meta-derived class prior, and class-count context. Exact features and preprocessing appear in Appendix B; no strong score, post-upgrade label, or evaluation outcome is included.

Paired supervision. Learning V requires paired Meta supervision, $D_M = \{(x_j, a_c(x_j), a_s(x_j), y_j)\}_{j=1}^{M}$, where y_j is a reference label used to form CC/WC/WW/CW outcomes. A four-class outcome model learns their conditional probabilities from cheap-side state. Reference labels are required for Meta supervision and offline evaluation, but are unavailable to the allocation policy on D_T. Meta is used to fit preprocessing and the outcome model; target examples contribute only cheap-side covariates at allocation time. In particular, the ranking cannot use a strong label or score, a post-upgrade outcome, or any Gold or Pool evaluation outcome.

AdaLabel is estimator-agnostic; our lightweight multinomial model isolates the estimand with limited predictive capacity, while any estimator of the four outcome probabilities can instantiate the same rule.

Paired Meta supervision is a substantive upfront cost excluded from per-target budgets. It is appropriate only when amortizable and when conditional upgrade behavior transfers under a stable annotator hierarchy (Appendix L).

Lemma 1 (Ranking equivalence under constant regression risk). Let $V_{\mathrm{SAV}}(x) = P(WC \mid s(x)) - P(CW \mid s(x))$ and $V_{\mathrm{WC}}(x) = P(WC \mid s(x))$. If $P(CW \mid s(x)) = c$ for all x, the rankings are identical since $V_{\mathrm{SAV}}(x) = V_{\mathrm{WC}}(x) - c$. For two untied candidates i, j, a strict reversal occurs iff

$$\big(P_i(WC) - P_j(WC)\big)\Big(\big[P_i(WC) - P_j(WC)\big] - \big[P_i(CW) - P_j(CW)\big]\Big) < 0.$$

Applicability and degeneracy. SAV applies to budgeted upgrades with a stronger but non-oracle annotator, where an upgrade may rescue or regress an existing annotation. By Lemma 1, constant $P(CW \mid s)$ makes SAV and WC-only rankings identical. More generally, SAV can differ from WC-only only when predicted regression risk changes candidate ordering; this does not guarantee higher realized utility, which also depends on outcome-model predictive quality and Meta-to-target transfer. AdaLabel therefore requires representative paired Meta supervision and optimizes direct annotation-correctness change, $U = N_{WC} - N_{CW}$, rather than guaranteed downstream improvement.

3.5 Budgeted Top-B allocation

AdaLabel sorts frozen scores, selects the first B entries using sample identifiers to break ties, and assigns

$$a_{final}(x_i) = \begin{cases} a_s(x_i), & \pi(x_i) = 1, \\ a_c(x_i), & \pi(x_i) = 0. \end{cases}$$

One ranking defines all budgets as nested prefixes; the downstream learner uses only the resulting labels.

Proposition 1 (Top-B optimality). Assume additive utility and a uniform strong-call cost. Conditional on the cheap-side states, a size-B set containing the B largest values $V(x)$ maximizes the conditional expected net utility, since $\mathbb{E}[U(S) \mid \{s_{cheap}(x)\}] = \sum_{x \in S} V(x)$. *Proof.* If a selected example has lower value than an unselected example, exchanging them increases this expectation by their positive value difference; repeating this exchange yields a Top-B set (with arbitrary tie resolution).

3.6 Utility and estimator implementation

For a selected set S, net annotation utility is

$$U(S) = \sum_{x \in S} \Delta Q(x) = \#WC(S) - \#CW(S).$$

The matched-budget effect is $U(S_{AdaLabel}) - U(S_{comparator})$ and evaluates labels, not downstream accuracy. AdaLabel trains a multinomial outcome model on Meta and ranks target examples using its cheap-only signed score; exact model, scaler, feature, class-order, and optimization details appear in Appendix B.

Direct MLP is the prespecified direct-value comparator; Direct Linear estimates the same target with matched features. WC-only, cheap-error, entropy, and margin provide alternative signals (Appendix B).

4 Experimental Setup

4.1 Data and annotators

The primary resource-aware evaluation uses disjoint Yahoo Answers Meta ($N = 2{,}000$) and Gold ($N = 4{,}000$) splits, with rankings frozen before Gold strong access and methods compared at matched logical-token budgets. Frozen Yahoo Answers and 20 Newsgroups records support attribution without new annotator calls, while Amazon Reviews Multi English provides an independent auditability replication. DBPedia14 (Zhang et al., 2015), Banking77 (Casanueva et al., 2020), TREC-6 (Li & Roth, 2002), and CLINC150 (Larson et al., 2019) provide supporting comparator, downstream, and transfer evidence. Full dataset, annotator, scoring, and split specifications appear in Appendix C.2.

4.2 Baselines and budgets

The DBPedia14 audit matches eligible population, strong annotator, and upgrade count. Learned comparators share AdaLabel's cheap-side features; scalar heuristics use their defining scores, and Oracle is descriptive only. Budgets of 10%, 20%, 30%, and 40% correspond to 402, 804, 1,206, and 1,608 upgrades; 20% and 40% are primary.

Table 1: Prospective budgeted annotation allocation on Yahoo Answers Gold ($N = 4{,}000$). Methods use frozen rankings and matched logical-token budgets. The effect is AdaLabel minus Direct MLP as a percentage of Gold; intervals use 10,000 paired reference-class-stratified bootstrap replicates. All four Holm-adjusted p values are < 0.001.

Budget	AdaLabel	Direct MLP	Direct Linear	WC-only	$\Delta U/N$	95% CI
10%	**239**	166	226	**239**	+1.83 pp	[0.90, 2.78]
20%	434	306	412	**438**	+3.20 pp	[2.02, 4.43]
30%	607	437	585	**609**	+4.25 pp	[2.92, 5.58]
40%	**755**	585	739	752	+4.25 pp	[2.88, 5.60]

Matched strong-call counts test whether different selections improve utility at the same number of upgrades. Yahoo Answers separately evaluates logical-token budgets and measured GPU time; the budget units and deployment-cost scope are detailed in Appendix J.

4.3 Metrics and evaluation protocol

The primary metric is net annotation utility $U(S)$, with one ranking supplying nested budget prefixes. The final DBPedia14 all-method comparison is a corrective post-Gold same-feature audit; Appendix J distinguishes it from the original prospectively frozen protocol.

Banking77 uses a frozen RoBERTa-base classifier trained on policy-constructed datasets at 20%, 30%, and 40% budgets, with matched calls and 15 seeds. Macro-F1 is primary; paired Student-t intervals (df$= 14$), 10,000 hierarchical bootstrap replicates (seed 20261121), and Holm correction across budgets are reported. TREC-6 and CLINC150 provide parser-aware transfer evidence, and DBPedia14 parser-free evidence; details appear in Appendices C and H.

5 Results

5.1 Stronger annotation creates rescues and regressions

The DBPedia14 Gold population contains 4,018 examples. The cheap annotator is correct on 50.15% and the strong annotator on 62.82%. Nevertheless, the upgrade outcomes include 870 WC rescues and 361 CW regressions (with 1,654 CC and 1,133 WW cases). Aggregate superiority of the strong annotator therefore does not imply sample-wise monotonic benefit. It identifies a positive population-level opportunity while leaving a nontrivial subset in which the upgrade lowers correctness.

The coexistence of these outcomes motivates the signed target. As supporting evidence, the corrective post-Gold same-feature DBPedia14 audit finds AdaLabel best among deployable policies at 10%, 30%, and 40%, while Cheap-error leads at 20%. Full matched-budget comparisons and rescue–regression decompositions are reported in Appendix D.

5.2 Resource-aware prospective evaluation

This experiment tests allocation under length-dependent annotation cost with an independent prospective design. We evaluated Yahoo Answers using disjoint Meta (2,000) and Gold (4,000) splits, with Qwen3-1.7B and Qwen3-8B as the cheap and strong annotators. BF16 teacher-forced scoring covered all ten candidates. Features, models, rankings, and nested selected sets were frozen before Gold strong outcomes were opened. Methods were compared at matched logical-token budgets using $U = N_{WC} - N_{CW}$.

AdaLabel exceeds the prespecified Direct MLP signed-value comparator by 73–170 net corrections under nearly identical token budgets; all four reference-class-stratified bootstrap comparisons remain significant after Holm adjustment (Table 1). Direct Linear and WC-only are competitive at some budgets, so the advantage is not uniform. The close WC-only values are consistent with Lemma 1: nearly constant predicted regression risk has little effect on ranking. Section 5.3 tests when heterogeneous, predictable risk changes allocation; the frozen Yahoo endpoint audit remains in Appendix M.1.

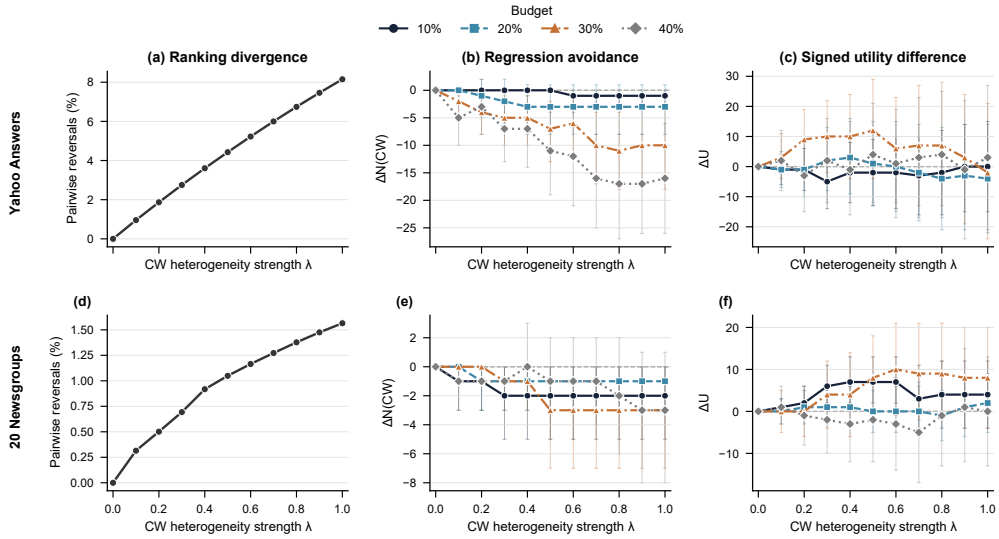

Figure 2: When signed value changes allocation. Exploratory centered frozen-record replay on Yahoo Answers (top) and 20 Newsgroups (bottom), from WC-only ($\lambda = 0$) to released SAV ($\lambda = 1$). Columns show ranking reversals, realized ΔN_{CW}, and ΔU, all relative to WC-only; negative ΔN_{CW} means fewer regressions. Error bars are nominal paired-bootstrap 95% intervals (10,000 replicates).

Across 24 synchronized CUDA-event replays, top-1 agreement with sealed Gold labels was 100%. At the complete 40% replay, AdaLabel and Direct MLP processed 2.763M and 2.761M logical tokens in 696.5 and 672.1 GPU seconds, respectively.

5.3 When does signed annotation value matter?

WC-only differs from SAV only through predicted regression risk. Thus, constant risk recovers the WC-only ranking, whereas heterogeneous risk can induce ranking divergence only when it is predictable from cheap-side information. An exploratory centered frozen-record replay on Yahoo Answers and 20 Newsgroups replaces that risk by

$$p_{i,\lambda}(CW) = \bar{p}_{CW} + \lambda\big[p_i(CW) - \bar{p}_{CW}\big], \qquad V_\lambda(x_i) = p_i(WC) - p_{i,\lambda}(CW),$$

for $\lambda \in [0, 1]$ and $\bar{p}_{CW} = N^{-1} \sum_i p_i(CW)$. This preserves mean risk while restoring sample heterogeneity, using only frozen probabilities and outcomes.

At $\lambda = 0$, both datasets exactly reproduce WC-only; at $\lambda = 1$, they exactly recover the released SAV rankings. Pairwise reversals reach 8.15% on Yahoo Answers and 1.56% on 20 Newsgroups (Figure 2).

Restoring heterogeneity generally reduces realized regressions, but ranking divergence does not imply higher realized utility at every budget because avoided CW cases can displace WC rescues. Frozen $P(CW \mid s)$ discriminates realized CW outcomes with AUROC 0.675 on Yahoo and 0.664 on 20 Newsgroups. Ranking divergence requires sufficiently heterogeneous predicted regression risk, while useful regression avoidance additionally requires that this risk align with realized CW outcomes (Appendix N).

5.4 Attributing signed annotation value

Having established prospective gains over the prespecified direct-value comparator under matched resource budgets, we next ask what information in the learned score accounts for those gains. In an exploratory frozen-record attribution analysis, we isolate three explanations using the Yahoo Answers and 20 Newsgroups records, without new annotator calls (Appendix P, Figure 18).

Four-state estimation exceeds the class-only signed prior by 90–223 utility units on Yahoo and 210–405 on 20 Newsgroups, with all nominal paired intervals above zero. Relative to matched three-state

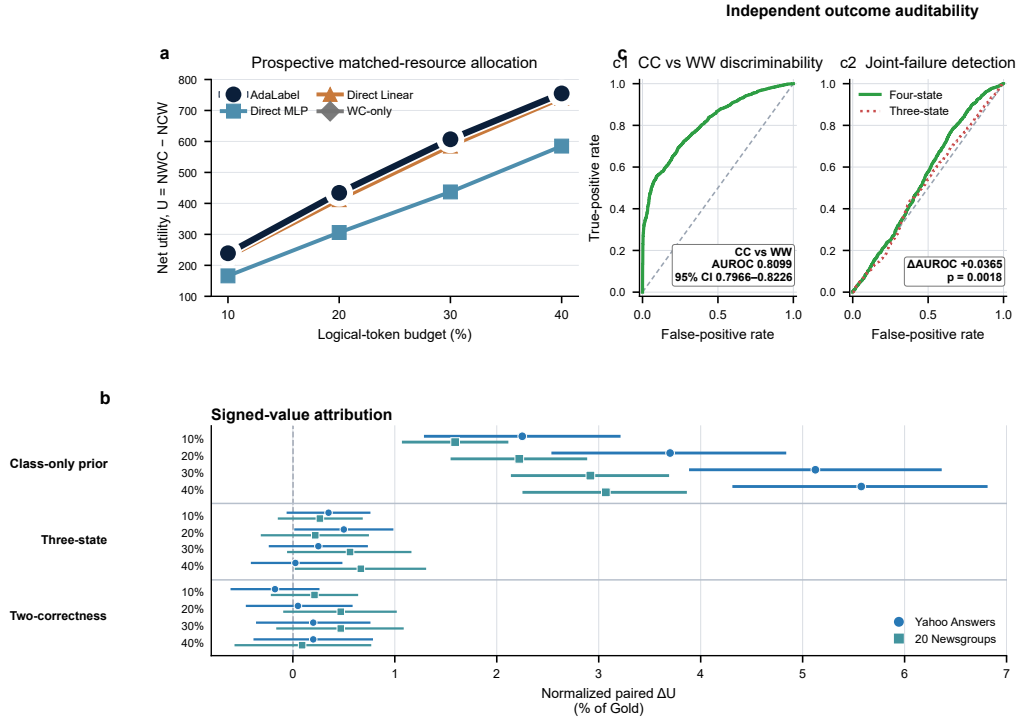

Figure 3: Prospective allocation, signed-value attribution, and outcome auditability. (a) Yahoo Answers allocation under prospectively frozen rankings and matched logical-token budgets. (b) Normalized paired utility differences between four-state AdaLabel and the class-only prior, matched three-state model, and two-correctness alternative on frozen Yahoo Answers and 20 Newsgroups records. Effects and exploratory nominal paired-bootstrap 95% intervals are percentage points of each Gold population; the vertical line marks no difference and intervals are not multiplicity-adjusted. (c) Independent Amazon auditability: (c1) Meta CC-versus-WW discrimination and (c2) held-out Gold joint-failure detection with four-state $P(WW)$ versus three-state $P(\text{ZERO})$.

Table 2: Outcome-representation ablation on frozen Yahoo Answers and 20 Newsgroups records. Higher $U = N_{WC} - N_{CW}$ is better.

Dataset	Method	10%	20%	30%	40%
Yahoo	Four-state	239	**434**	**607**	**755**
	Three-state	225	414	597	754
	Two-correctness	**246**	432	599	747
	Class-prior only	149	286	402	532
20 Newsgroups	Four-state	**800**	**1484**	**2156**	**2761**
	Three-state	765	1455	2082	2673
	Two-correctness	772	1422	2094	2749
	Class-prior only	590	1191	1771	2356

estimation, point estimates are positive at all eight dataset–budget cells, but nominal intervals exclude zero only at selected operating points. Every four-state–minus–two-correctness interval includes zero, so four-state is competitive rather than uniquely superior as a parameterization of the signed target. Table 2 reports the observed raw utilities, while Figure 3(b) shows the corresponding normalized paired effects and nominal bootstrap intervals.

Table 3: Parser-aware held-out transfer effects (AdaLabel minus Direct MLP).

Dataset	20% ΔU [95% CI]	40% ΔU [95% CI]
TREC-6	+18 [6, 31]	+17 [5, 29]
CLINC150	+6 [−9, 25]	+19 [−4, 48]

5.5 Four-state outcome auditability

In a prospectively frozen Amazon replication, CC versus WW discrimination reached AUROC 0.8099 (95% CI [0.7966, 0.8226]). On held-out Gold, four-state $P(WW)$ achieved AUROC 0.5530 versus 0.5165 for three-state $P(\mathrm{ZERO})$ (difference 0.0365, $p = 0.0018$); the shared WC/CW Brier comparison was not significant. These results support auditability, not universal utility or calibration superiority.

5.6 Downstream consequences and transfer stress tests

On Banking77, AdaLabel improves mean Macro-F1 over Direct MLP by 0.64, 0.36, and 1.18 points at 20%, 30%, and 40%; only the 40% effect survives Holm correction ($p = 0.019$). Table 3 makes the heterogeneous transfer evidence explicit: TREC-6 intervals exclude zero, whereas CLINC150 intervals do not.

Across the prespecified TREC-6 hierarchies, rescues and regressions coexist; together with the supplementary 20 Newsgroups A/B study, this supports heterogeneous rather than universal transfer (Appendices G, E, F, and O).

6 Discussion

Error detection and upgrade valuation differ: a likely cheap-label error may remain wrong after replacement, while a correct label can regress. Cheap-error combines correctable WC with WW failures; SAV evaluates rescue minus regression. WC-only ranks rescue probability but omits the harmful CW term.

SAV differs from WC-only only under heterogeneous, predictable regression risk. Constant $P(CW \mid s)$ leaves their rankings identical, whereas predicted sample-dependent risk can reorder candidates. Ranking divergence reflects learnable CW variation, not evidence of improvement at every budget.

Avoiding CW cases can displace WC rescues ($\Delta U = \Delta N_{WC} - \Delta N_{CW}$), so fewer regressions need not yield higher utility. Amazon supports four-state auditability, but neutral Brier and heterogeneous utility results bound that claim. AdaLabel is most appropriate when risk is predictable and paired Meta supervision transfers.

7 Limitations

AdaLabel requires representative paired Meta annotations and reference labels, with upfront cost and sensitivity to distribution shift; rankings also depend on the annotator pair, prompt, decoding protocol, and reference standard.

Matched strong-call and logical-token budgets do not equalize monetary price or heterogeneous deployment cost. Net annotation utility is not guaranteed downstream gain, and the scope is classification; structured, multilabel, multimodal, and reference-free tasks may require other utilities (Appendix L).

8 Conclusion

SAV targets expected rescue minus regression. It can change allocation only under predictable variation in regression risk, and fewer regressions can displace rescues. The heterogeneous evidence supports modeling upgrade consequences rather than assuming uniformly beneficial replacement.

AI USE STATEMENT

AI-assisted tools were used only to improve the language and presentation of the manuscript. The authors checked all scientific claims and numerical values against the recorded materials and remain responsible for the final manuscript. No AI system generated observations or altered frozen results.

REPRODUCIBILITY STATEMENT

The anonymous supplementary package includes the all-method utility table, Direct Linear and Direct MLP comparator specifications, AdaLabel outcome model and preprocessing, DBPedia14 split definitions, rankings, outcome tables, and bootstrap records needed to reproduce the reported matched-budget comparisons. The original Gold protocol fixed rankings before Gold strong-outcome access; the final DBPedia14 all-method table is a corrective post-Gold audit (Appendix J). The Yahoo package additionally includes its frozen split index, candidate scores, features, token costs, selected sets, bootstrap records, 24 replay traces, and an offline numerical verifier.

REFERENCES

Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S. Weld. Is the most accurate AI the best teammate? optimizing AI for teamwork. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11405–11414, 2021. doi: 10.1609/aaai.v35i13.17359.

Peter L. Bartlett and Marten H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(59):1823–1840, 2008. URL https://jmlr.org/papers/v9/bartlett08a.html.

Tolga Bolukbasi, Joseph Wang, Ofer Dekel, and Venkatesh Saligrama. Adaptive neural networks for efficient inference. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 527–536. PMLR, 2017. URL https://proceedings.mlr.press/v70/bolukbasi17a.html.

Inigo Casanueva, Tomas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. Efficient intent detection with dual sentence encoders. In *Proceedings of INTERSPEECH*, 2020.

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406.

David A. Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994. doi: 10.1007/BF00993277.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996. doi: 10.1613/jair.295.

Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. Learning with rejection. In *Proceedings of the 27th International Conference on Algorithmic Learning Theory*, pp. 67–82. Springer, 2016. doi: 10.1007/978-3-319-46379-7_5.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979. doi: 10.2307/2346806.

Abir De, Nastaran Okati, Ali Zarezade, and Manuel Gomez-Rodriguez. Classification under human assistance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 5905–5913, 2021. doi: 10.1609/aaai.v35i7.16738.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 11173–11195. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.626. URL https://aclanthology.org/2023.acl-long.626/.

Ran El-Yaniv and Yair Wiener. On the foundations of noise-free selective classification. *Journal of Machine Learning Research*, 11(53):1605–1641, 2010. URL https://jmlr.org/papers/v11/el-yaniv10a.html.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1183–1192. PMLR, 2017. URL https://proceedings.mlr.press/v70/gal17a.html.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems 30*, pp. 4878–4887, 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/4a8423d5e91fda00bb7e46540e2b0cf1-Abstract.html.

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*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2242–2251. PMLR, 2019. URL https://proceedings.mlr.press/v97/ghorbani19c.html.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi: 10.1073/pnas.2305016120.

Alex Graves. Adaptive computation time for recurrent neural networks, 2016. URL https://arxiv.org/abs/1603.08983.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. AnnoLLM: Making large language models to be better crowdsourced annotators. In *Proceedings of NAACL-HLT 2024, Industry Track*, pp. 165–190. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-industry.15. URL https://aclanthology.org/2024.naacl-industry.15/.

Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3301–3310. PMLR, 2019. URL https://proceedings.mlr.press/v97/kaya19a.html.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1885–1894. PMLR, 2017. URL https://proceedings.mlr.press/v70/koh17a.html.

Stefan Larson, Elizabeth Mahoney, Erin Leach, Andrew Cao, Chen Zhang, and Aravind Krishnamurthy. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of EMNLP-IJCNLP*, pp. 5020–5025, 2019.

Dong-Ho Lee, Jay Pujara, Mohit Sewak, Ryen White, and Sujay Jauhar. Making large language models better data creators. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 15349–15360. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.948. URL https://aclanthology.org/2023.emnlp-main.948/.

David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pp. 3–12. ACM/Springer, 1994. doi: 10.1007/978-1-4471-2099-5_1.

Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy Chen, Zhengyuan Liu, and Diyi Yang. CoAnnotating: Uncertainty-guided work allocation between human and large language models for data annotation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1487–1505. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.92. URL https://aclanthology.org/2023.emnlp-main.92/.

Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of COLING*, 2002.

David J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992. doi: 10.1162/neco.1992.4.4.590.

David Madras, Toniann Pitassi, and Richard S. Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Advances in Neural Information Processing Systems 31*, pp. 6150–6160, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/09d37c08f7b129e96277388757530c72-Abstract.html.

Andrew McCallum and Kamal Nigam. Employing EM and pool-based active learning for text classification. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 350–358. Morgan Kaufmann, 1998.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7076–7087. PMLR, 2020. URL https://proceedings.mlr.press/v119/mozannar20b.html.

Hussein Mozannar, Hunter Lang, Dennis Wei, Prasanna Sattigeri, Subhro Das, and David Sontag. Who should predict? exact algorithms for learning to defer to humans. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 10520–10545. PMLR, 2023. URL https://proceedings.mlr.press/v206/mozannar23a.html.

Curtis G. Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/f2217062e9a397a1dca429e7d70bc6ca-Abstract-round1.html.

Nastaran Okati, Abir De, and Manuel Gomez-Rodriguez. Differentiable learning under triage. In *Advances in Neural Information Processing Systems 34*, pp. 9140–9151, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/4c4c937b67cc8d785cea1e42ccea185c-Abstract.html.

Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason A. Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. *The VLDB Journal*, 29(2–3): 709–730, 2020. doi: 10.1007/s00778-019-00552-1.

Alexander J. Ratner, Christopher M. De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems 29*, pp. 3567–3575, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/6709e8d64a5f47269ed5cea9f625f7ab-Abstract.html.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(43): 1297–1322, 2010. URL https://jmlr.org/papers/v11/raykar10a.html.

Filipe Rodrigues and Francisco C. Pereira. Deep learning from crowds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 1611–1618, 2018. doi: 10.1609/aaai.v32i1.11506.

Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the 18th International Conference on Machine Learning*, pp. 441–448. Morgan Kaufmann, 2001.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1aIuk-RW.

H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pp. 287–294. ACM, 1992. doi: 10.1145/130385.130417.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Victor S. Sheng, Foster Provost, and Panagiotis G. Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 614–622. ACM, 2008. doi: 10.1145/1401890.1401965.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. Cheap and fast—but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pp. 254–263. Association for Computational Linguistics, 2008. URL https://aclanthology.org/D08-1027/.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 930–957. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.54. URL https://aclanthology.org/2024.emnlp-main.54/.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. BranchyNet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition*, pp. 2464–2469. IEEE, 2016. doi: 10.1109/ICPR.2016.7900006.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2001. URL https://jmlr.org/papers/v2/tong01a.html.

Rajeev Verma and Eric Nalisnick. Calibrated learning to defer with one-vs-all classifiers. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 22184–22202. PMLR, 2022. URL https://proceedings.mlr.press/v162/verma22c.html.

Rajeev Verma, Daniel Barrejon, and Eric Nalisnick. Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 11415–11434. PMLR, 2023. URL https://proceedings.mlr.press/v206/verma23a.html.

Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. I–511–I–518. IEEE, 2001. doi: 10.1109/CVPR.2001.990517.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier R. Movellan, and Paul L. Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in Neural Information Processing Systems 22*, pp. 2035–2043, 2009. URL https://proceedings.neurips.cc/paper/2009/hash/f899139df5e1059396431415e770c6dd-Abstract.html.

Yair Wiener and Ran El-Yaniv. Agnostic selective classification. In *Advances in Neural Information Processing Systems 24*, pp. 1665–1673, 2011. URL https://proceedings.neurips.cc/paper/2011/hash/4b6538a44a1dfdc2b83477cd76dee98e-Abstract.html.

Bryan Wilder, Eric Horvitz, and Ece Kamar. Learning to complement humans. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 1526–1533, 2020. doi: 10.24963/ijcai.2020/212.

Haodi Zhang, Junyu Yang, Jinyin Nie, Peirou Liang, Kaishun Wu, Defu Lian, Rui Mao, and Yuanfeng Song. Efficient data labeling by hierarchical crowdsourcing with large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 11290–11303. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.748/.

Ruoyu Zhang, Yanzeng Li, Yongliang Ma, Ming Zhou, and Lei Zou. LLMaAA: Making large language models as active annotators. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 13088–13103. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.872. URL https://aclanthology.org/2023.findings-emnlp.872/.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, 2015.

A Extended Method Details

A.1 Interpretation of the conditional target

The conditional expectation is defined with respect to information available before an upgrade. In particular, $s_{cheap}(x)$ may contain the cheap annotator's class-score distribution and derived summaries, but it cannot contain $a_s(x)$, a strong score, or any outcome computed after a strong call. This temporal restriction gives $V(x)$ a prospective interpretation: it is the value predicted when the decision is made rather than a relabeling of an observed strong outcome. Meta outcomes identify the conditional relationship, while the allocation population supplies only covariates at decision time.

The four outcome classes make the estimand auditable. Their probabilities sum to one, and the zero coefficients for CC and WW ensure that examples whose correctness would not change do not receive positive value merely because they are difficult. Conversely, a high probability of CW reduces the score even when the probability of WC is also high. A policy can therefore prefer a moderately uncertain example with a favorable rescue-to-regression balance over a more uncertain example whose strong annotation is risky.

A.2 Estimation and separation of roles

The Meta feature matrix is standardized using parameters fit on Meta only. The outcome target is represented as a four-class categorical variable, and the fitted predictor returns a probability vector in the fixed class order. The scalar ranking score is derived after prediction by the signed contrast $P(WC \mid s_{cheap}) - P(CW \mid s_{cheap})$. This separation between probability estimation and scalar utility keeps the score transparent: changing a utility convention would change the contrast, whereas the outcome model remains an interpretable distribution over annotation outcomes.

No Pool or Gold label is used to select features, fit preprocessing, choose the model family, or tune a ranking threshold. Pool and Gold are scored only after the value model is fixed. In the original Gold protocol and the independent prospective evaluations, the complete ranking is fixed before strong outcomes are opened, and the same ranking supplies every budget. The corrective DBPedia14 same-feature audit keeps the Gold population, outcomes, budgets, ranking rule, estimator configurations, and baseline set fixed. The 10%, 20%, 30%, and 40% rows are nested prefixes of one decision rule rather than outcome-driven searches.

A.3 Relation to alternative objectives

If the strong annotator were assumed perfect, cheap-error probability could serve as a proxy for upgrade value. The CC/WC/WW/CW formulation removes that assumption. It also differs from maximizing disagreement: disagreement includes both WC and CW, whereas the scientific objective

assigns opposite signs to them. Similarly, ranking by low cheap confidence or high entropy does not encode whether the strong annotator is likely to correct or overturn the cheap label. Such quantities can be useful baselines, but they are not algebraically equivalent to $V(x)$.

A.4 From scores to a constructed dataset

After the selected set S is identified, the annotation pipeline requests strong labels only for S. Cheap labels are retained for $D \setminus S$. The resulting dataset can be passed to any downstream learner without exposing the value model or outcome taxonomy. AdaLabel controls which examples are upgraded, while the downstream learner determines how the resulting labels affect model parameters. This modularity permits evaluation through both direct label utility $U(S)$ and a separately trained classifier.

A.5 Scope and assumptions

The method assumes that the cheap/strong hierarchy is fixed during an evaluation and that reference labels are available for Meta supervision and evaluation. It also assumes enough paired Meta examples to estimate outcome probabilities that transfer to the allocation population. These assumptions are stronger than those of an uncertainty-only heuristic and define the applicability of AdaLabel. They do not imply that the strong annotator is globally better, only that its conditional upgrade behavior can be estimated for the task at hand.

B Full Feature Specification

Let z_j be the mean teacher-forced log-probability assigned to candidate class j, and let $p_j = \exp(z_j)/\sum_{\ell=1}^{C}\exp(z_\ell)$. For DBPedia14, $C = 14$. The frozen ten-dimensional feature vector is ordered as

$$\phi(x) = \Bigg[-\frac{\sum_j p_j \log(p_j + 10^{-10})}{\log C},\ p_{(1)},\ p_{(2)},\ p_{(1)} - p_{(2)},\ \sum_{j=1}^{\min(5,C)} p_{(j)},$$
$$\mathrm{std}_{\mathrm{pop}}(z),\ \max_j z_j - \min_j z_j,\ \frac{\mathrm{LSE}(z)}{C},\ V_{\mathrm{class}},\ \log C \Bigg],$$

where $(1), (2), \ldots$ index probabilities in descending order and $\mathrm{LSE}(z) = \log\sum_j \exp(z_j)$. The standard deviation uses the population convention. The supplementary file `experiments/phase_a/FINAL/manifests/feature_schema.json` records this order and its frozen hashes.

For cheap-predicted class k, the frozen prior is

$$V_{\mathrm{class}}(k) = \frac{N_{WC,k} - N_{CW,k}}{N_k + 1}.$$

Thus the additive denominator smoothing constant is $\alpha = 1$. An unseen class uses the corresponding training subset's global value $(N_{WC} - N_{CW})/(N + 1)$. For Meta rows, V_{class} is cross-fitted: each validation fold receives a value computed only from the other four Meta folds, using the frozen KFold assignments. For Pool and Gold rows, V_{class} is computed from the full 3,990-row Meta outcome table. No Pool or Gold strong outcome, post-allocation label, or post-upgrade state is included in the allocation features.

The outcome model is multinomial logistic regression with L2 penalty, $C = 1.0$, LBFGS, `max_iter=2000`, no class weighting, and `random_state=20261001`. Probabilities are accessed by their semantic class labels rather than a presumed column order. A single StandardScaler is fit on the corrective Meta OOF matrix and then applied unchanged to Meta, Pool, and Gold. AdaLabel scores $P(WC \mid s_{cheap}) - P(CW \mid s_{cheap})$.

Direct Linear is ridge regression with $\alpha = 1.0$. Direct MLP is an MLPRegressor with one eight-unit ReLU hidden layer, Adam, learning rate 10^{-3}, L2 coefficient 10^{-3}, `max_iter=500`, and `random_state=20261001`. It uses early stopping with a 0.1 validation fraction and 20 non-improving iterations. Both direct models use squared-error regression on the same signed target $\Delta Q \in \{-1, 0, +1\}$, with WC$= +1$, CW$= -1$, and CC/WW$= 0$. Their comparison

with AdaLabel therefore evaluates different estimators of the same signed utility target. WC-only reuses the multinomial model's $P(WC)$, while Cheap-error uses $P(WC) + P(WW)$. Entropy is ranked descending, margin is ranked ascending through its negative, and every tie is broken by ascending `sample_id`. Full estimator details and the exact verification entry point are provided in `experiments/phase_a/FINAL/README.md`, `reproducibility/reproduce_phase_a.md`, and `scripts/verification/verify_phase_a.py` of the anonymous supplement.

C Additional Experimental Protocol

C.1 Dataset roles and interfaces

The evidence spans Banking77, TREC-6, CLINC150, and DBPedia14 with dataset-specific splits and annotation interfaces. DBPedia14 contains 3,990 Meta examples, 11,984 Pool examples, and 4,018 Gold examples. Meta provides paired cheap/strong outcomes for learning. Pool supports validation before the final held-out stage. Gold supplies the held-out parser-free evaluation population. This partitioning separates outcome-model fitting, pre-Gold validation, the original Gold ranking freeze, and the later corrective same-feature audit.

TREC-6 and CLINC150 use parser-aware evaluation and therefore provide complementary but interface-dependent evidence. DBPedia14 evaluates frozen legal class verbalizers with teacher-forced candidate scoring, avoiding free-form response parsing. Banking77 supplies the downstream classifier experiment used to test whether policy-dependent annotations are reflected after model training.

C.2 Annotator Hierarchy and Candidate Scoring

For DBPedia14, the frozen cheap annotator is Qwen3-1.7B (model and tokenizer revision `70d244cc86ccca08cf5af4e1e306ecf908b1ad5e`); the frozen strong annotator is Qwen3-8B (model and tokenizer revision `b968826d9c46dd6066d109eabc6255188de91218`). Both use prompt version `annotation_p8_v1`, parser version `label_parser_v1`, temperature 0, seed 20260830, deterministic decoding (`do_sample=false`, `enable_thinking=false`), and 16 maximum new tokens. The cheap scorer uses float16; the strong scorer uses bfloat16; neither uses quantization. For DBPedia14, candidate labels are the 14 canonical class names in `protocols/verbalizers.json`. Each candidate is appended after `Text:` <text> followed by a newline and `Class:` with a leading space and scored by the mean teacher-forced token log-probability over the candidate tokens. Candidate probabilities are the softmax of these length-normalized scores (an uncalibrated ranking distribution), and the predicted label is the maximum-score candidate. These settings define the DBPedia14 hierarchy. Dataset-specific departures, including Yahoo Answers precision and the Amazon hierarchy, are described in their respective appendices.

C.3 Annotators and allocation state

Each example is first scored by a cheap annotator. A strong annotator is an optional upgrade, and both outputs are compared with the same reference label. The strong annotator is never treated as an oracle. Allocation state is computed before any strong call from cheap-side probability and score summaries, the Meta-derived class-conditioned statistic, and task context. A paired Meta set is required to construct the four outcome targets and fit the value model.

C.4 Baselines and budgets

The Phase A audit compares AdaLabel with Direct Linear, Direct MLP, WC-only, Cheap-error, Entropy, Margin, Random, and Oracle. Direct Linear and Direct MLP use the same full eligible feature representation as AdaLabel. Entropy, Margin, WC-only, and Cheap-error are scalar policies by definition. Every deployable method is evaluated on the same eligible examples, with the same strong annotator and the same number of upgrades. Oracle is reported only as a descriptive non-deployable upper bound. Budget fractions of 10%, 20%, 30%, and 40% correspond to 402, 804, 1,206, and 1,608 upgrades on DBPedia14 Gold.

Table 4: Final Phase A all-method utility on DBPedia14 Gold. Column headers include the matched selected count.

Method	10% (402)	20% (804)	30% (1,206)	40% (1,608)
Direct Linear	169	294	392	498
Direct MLP	117	250	339	408
WC-only	136	247	381	529
Cheap-error	163	**315**	427	551
Entropy	147	254	357	456
Margin	138	266	400	495
AdaLabel	**178**	302	**436**	**568**
Oracle	402	804	870	870
Random	50.9	102.0	153.0	203.9

C.5 Downstream protocol

Banking77 uses the official data with frozen sample identifiers. The annotation pool contains 8,127 rows, of which 7,588 satisfy the frozen parser-clean eligibility rule. The Meta validation split contains 1,876 gold-labeled rows, and the fixed Gold test split contains 3,080 rows. At 20%, 30%, and 40%, each policy upgrades exactly 1,518, 2,277, and 3,036 of the same 7,588 training examples. All other examples retain their cheap labels.

Every constructed dataset trains the same 77-class `FacebookAI/roberta-base` model at revision `e2da8e2f811d1448a5b465c236feacd80ffbac7b`. Inputs are truncated or padded to 128 tokens. Training uses AdamW with learning rate 2×10^{-5}, weight decay 0.01 excluding bias and LayerNorm weights, a constant schedule without warmup, batch size 32, evaluation batch size 64, BF16, gradient clipping at 1.0, and at most ten epochs. Meta Macro-F1 selects the checkpoint with patience two; ties retain the first strictly best epoch. Training loaders shuffle with a seed-specific generator, whereas validation and test loaders do not shuffle. Deterministic PyTorch algorithms are enabled. The Gold test split is evaluated only after Meta checkpoint selection. The two policies share the data splits, selected counts, classifier initialization rule, optimization settings, and fifteen seed identifiers; only the constructed training labels differ.

The anonymous supplement retains the aggregate Banking77 source table in `experiments/phase_b/FINAL/results/aggregate_results.csv`. Per-seed run records, checkpoints, and training-freeze artifacts are intentionally excluded from the reviewer package. Annotation and downstream conditions are coupled, so the experiment is descriptive rather than an independent causal test of the DBPedia14 result.

D Final Phase A Same-Feature Audit

The final Phase A appendix reports the results of the same-feature fairness evaluation. The DBPedia14 Gold budget curve contains four fixed prefixes with 402, 804, 1,206, and 1,608 selected examples. AdaLabel is the highest deployable method at 10%, 30%, and 40%, while Cheap-error is higher at 20% by 13 utility units.

Figure 4 plots the same DBPedia14 Gold utility values as Table 4, making the budget-dependent ordering visible. AdaLabel is the best deployable policy at 10%, 30%, and 40%; the single exception is 20%, where Cheap-error attains 315 against AdaLabel's 302. Oracle is not a deployable policy and therefore does not participate in the ordering, and the Random band is descriptive rather than inferential.

The decomposition shows why allocation policies diverge in utility. AdaLabel's advantage does not come from finding more cheap errors in the abstract, but from the joint action of rescue concentration and regression avoidance. At 40%, AdaLabel attains the largest selected N_{WC}, 583, while keeping

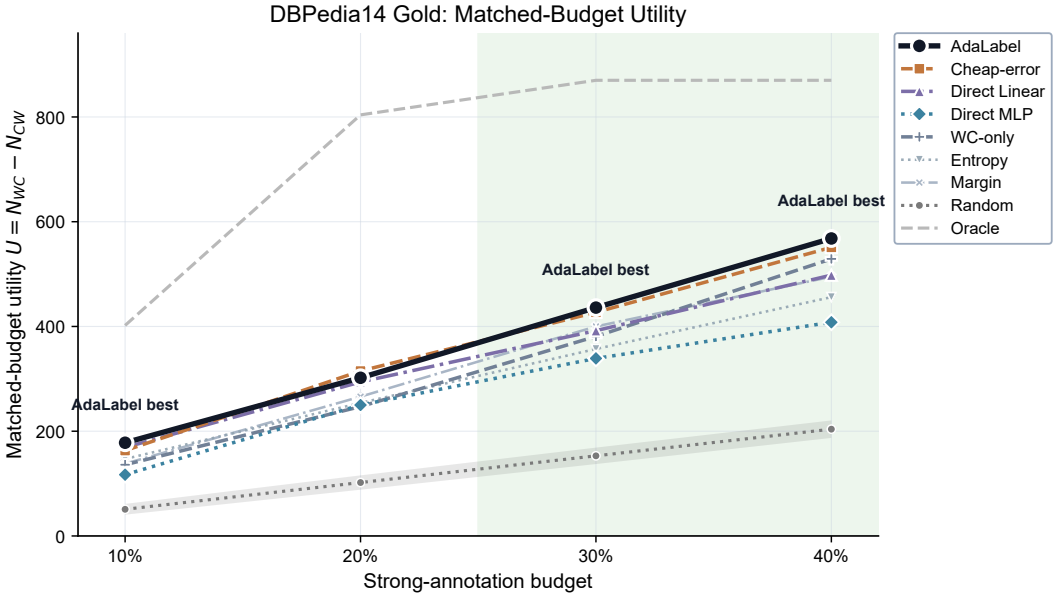

Figure 4: DBPedia14 Gold matched-budget utility $U = N_{WC} - N_{CW}$ (numerically identical to Table 4). Each curve is one allocation policy evaluated on the same frozen Gold population at identical upgrade counts. AdaLabel is drawn as the heavy black curve; Cheap-error, Direct Linear, Direct MLP, and WC-only are primary comparators; Entropy and Margin are secondary baselines; the Random reference shows its mean with a shaded mean $\pm$ SD band; and Oracle is a dashed non-deployable upper bound for reference only. Labels mark the budgets at which AdaLabel is the best deployable policy (10%, 30%, 40%); at 20% the best deployable policy is Cheap-error (315 versus 302), so no label appears.

Table 5: Detailed rescue–regression decomposition under matched Gold budgets. Entries are selected-outcome counts N_{WC} (rescues) and N_{CW} (regressions); net utility is $U = N_{WC} - N_{CW}$. The 20% and 40% budgets are primary. Cheap-error leads at 20% (315 versus 302).

Method	N_{WC}@20	N_{CW}@20	U@20	N_{WC}@40	N_{CW}@40	U@40
AdaLabel	311	9	302	583	15	568
Direct Linear	302	8	294	516	18	498
Direct MLP	274	24	250	465	57	408
WC-only	256	9	247	544	15	529
Cheap-error	321	6	315	574	23	551
Entropy	283	29	254	510	54	456
Margin	306	40	266	572	77	495
Oracle	804	0	804	870	0	870
Random	174.3	72.3	102.0	348.5	144.6	203.9

N_{CW} at 15. At 20%, Cheap-error selects more rescues, 321 versus 311, and fewer regressions, 6 versus 9, which explains its 13-unit advantage. WC-only matches AdaLabel's N_{CW} at both reported primary budgets but accumulates fewer rescues. AdaLabel is therefore the best deployable policy at three of four budgets without dominating every operating point.

Figure 5 visualizes the same decomposition.

For archival completeness, Figure 6 preserves the earlier composite visualization. Its DBPedia14 corrective-audit panels and independent Amazon auditability panels remain unchanged; the main text instead uses Figure 3 to reflect the revised evidence hierarchy.

Figure 7 visualizes this ablation.

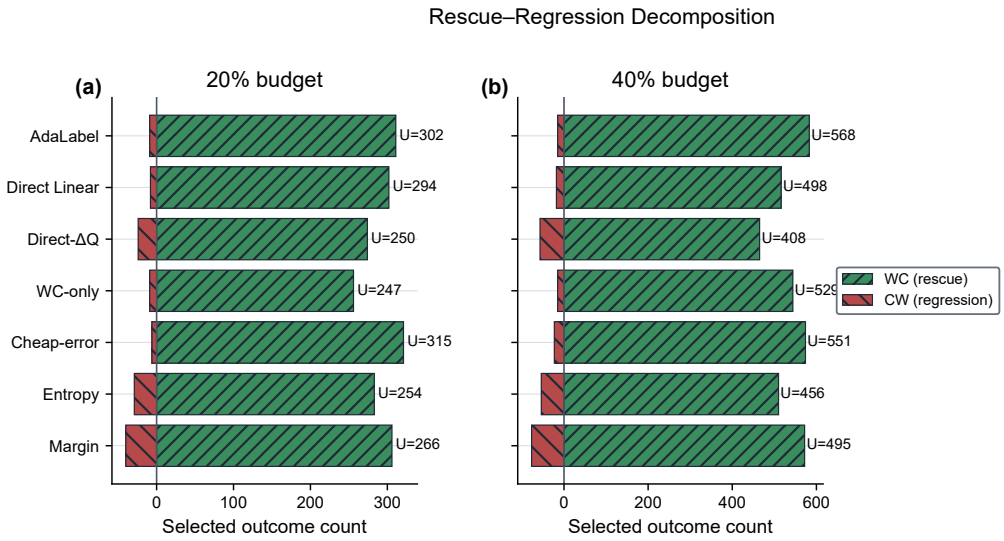

Figure 5: Rescue–regression decomposition under matched DBPedia14 Gold budgets. Direct-ΔQ denotes Direct MLP. The figure visualizes Table 5. Green hatched bars show selected WC rescues to the right; red hatched bars show selected CW regressions to the left; labels report $U = N_{WC} - N_{CW}$. The 20% and 40% panels make the signed allocation objective explicit.

Table 6: Final Phase A feature ablation for V_{class}. The ablation removes only the V_{class} column from the shared preprocessed representation.

Budget	Full WC	Full CW	Full U	w/o WC	w/o CW	w/o U	ΔU
10%	183	5	178	112	17	95	83
20%	311	9	302	217	40	177	125
30%	448	12	436	354	52	302	134
40%	583	15	568	495	69	426	142

Figure 8 shows the full reference distribution behind these summary statistics.

For every AdaLabel row, the reported utility equals $N_{WC} - N_{CW}$: $183 - 5 = 178$, $311 - 9 = 302$, $448 - 12 = 436$, and $583 - 15 = 568$. The population net rescue is 509, but selected-set utility may exceed this total because a policy can leave negative-contribution CW cases unupgraded.

E Cross-Dataset Results

Cross-dataset results are reported separately rather than pooled into a single claim. TREC-6 has parser-aware effects of +18 at 20% (95% CI [6,31]) and +17 at 40% ([5,29]), with both intervals excluding zero. CLINC150 has intervals that include zero, with +6 at 20% ([-9,25]) and +19 at 40% ([-4,48]). DBPedia14 provides parser-free Gold evidence through the corrective post-Gold same-feature audit.

Parser-aware and parser-free interfaces should not be treated as interchangeable replications. Free-form parsing can turn a representational preference into an apparent annotation error. DBPedia14 directly scores legal candidate labels, while TREC-6 and CLINC150 remain interface-dependent stress tests.

Figure 9 plots the TREC-6 and CLINC150 differences reported in Table 3; DBPedia14 remains separate because it uses a parser-free interface and a corrective audit design.

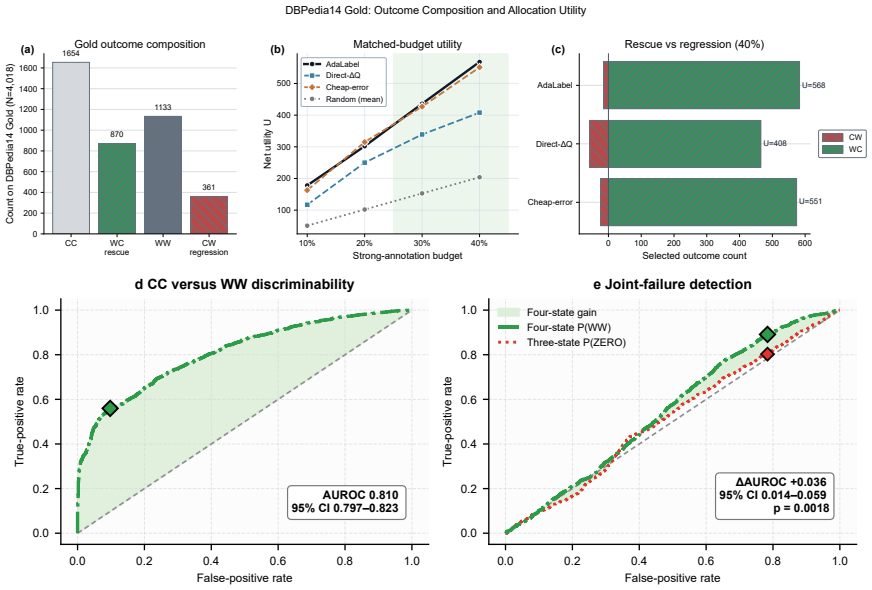

Figure 6: Archived composite visualization of the DBPedia14 corrective audit and Amazon outcome auditability. Direct-ΔQ in the figure denotes Direct MLP. The upper row reports the DBPedia14 Gold audit: outcome composition across CC, WC, WW, and CW on $N = 4{,}018$ frozen Gold examples; matched-budget net utility $U = N_{WC} - N_{CW}$ for deployable policies under corrected same-feature evidence; and rescue–regression decomposition at the 40% budget. The lower row reports the independent Amazon Reviews Multi English replication: Meta-only five-fold out-of-fold discrimination between CC and WW, and held-out Gold joint-failure detection with four-state $P(WW)$ versus three-state $P(\text{ZERO})$. The latter has a paired AUROC difference of 0.0365 (95% CI [0.0137, 0.0589], $p = 0.0018$).

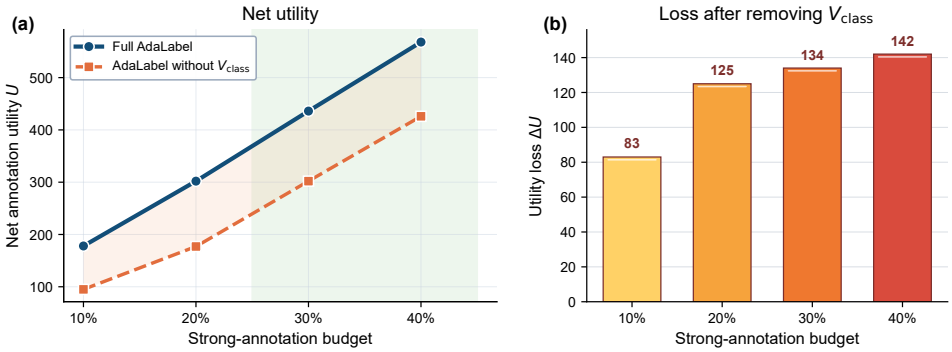

Figure 7: Feature ablation for V_{class} on DBPedia14 Gold, visualizing Table 6. Panel (a) compares the full AdaLabel representation against the version with only V_{class} removed from the shared preprocessing pipeline; panel (b) reports the corresponding utility loss.

F Annotator-Hierarchy Robustness on TREC-6

The three hierarchies below were pre-specified before any held-out outcome was inspected, and all results use the frozen held-out TREC-6 population (500 examples) at matched budgets of 20%, 30%, and 40%. They extend the allocation study across multiple cheap–strong gaps within the Qwen3 family. They provide additional robustness evidence rather than a new Phase A or Phase B claim.

Every hierarchy shows WC>0 and CW>0, so the stronger annotator is non-oracle: it both rescues and regresses. Under H3 (1.7B$\rightarrow$8B), the aggregate accuracy gap is only 4.4 percentage points, yet the annotators produce 90 rescues and 68 regressions. Aggregate annotator quality

Table 7: Random baseline distribution over 10,000 rankings. Statistics summarize net utility U.

Budget	Mean	SD	2.5%	Median	97.5%
10%	50.9	10.3	31	51	71
20%	102.0	13.7	76	102	129
30%	153.0	15.7	122	153	184
40%	203.9	16.7	171	204	237

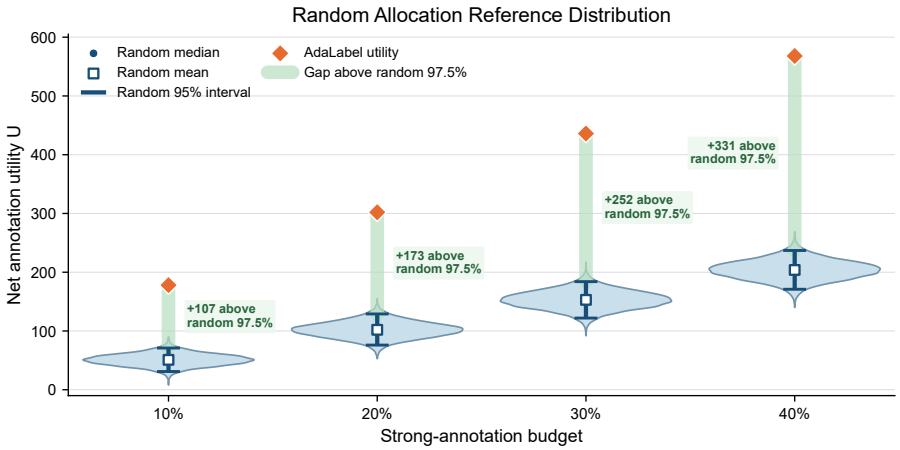

Figure 8: Distribution of net utility over 10,000 frozen random rankings at each DBPedia14 Gold budget, visualizing Table 7. Violin shapes summarize the stored random utilities rather than simulated samples; vertical segments show the reported 2.5–97.5% range, circles the empirical median, and squares the source mean. Diamonds show the corresponding AdaLabel utility. This is a descriptive random-allocation reference, not a fitted uncertainty model.

therefore does not determine instance-level upgrade value, which is captured by the signed margin $P(\text{WC} \mid s) - P(\text{CW} \mid s)$.

Figure 10 visualizes this characterization.

AdaLabel exceeds random allocation at all nine hierarchy–budget cells, but it is not the unique deployable leader: Cheap-error attains the highest utility at H1/40%, Entropy leads H3/30%, Direct MLP leads H3/40%, Direct Linear leads H1/30%, and Direct MLP and Margin tie at H3/20%. Observed utility is substantially larger in H1 and H2, which use the weaker 0.6B cheap annotator. In H3, the 1.7B cheap annotator already reaches 59.4% accuracy. This comparison describes the evaluated settings; it does not establish monotone scaling with annotator weakness. The study is confined to TREC-6 with three pre-specified hierarchies and should be read as additional annotator-robustness evidence, not as a universal or uniformly dominant performance result.

Figure 11 plots the corresponding matched-budget utility curves.

G Final Phase B Downstream Results

The Banking77 downstream experiment trains RoBERTa-base classifiers on datasets constructed under AdaLabel and the Direct MLP comparator at 20%, 30%, and 40% strong-annotation budgets. Both policies receive the same number of strong calls at each budget. The final study uses fifteen frozen downstream training seeds (20260910, 20260911, 20260912, 20260913, 20260914, 20261101, 20261102, 20261103, 20261104, 20261105, 20261106, 20261107, 20261108, 20261109, 20261110). Macro-F1 is the primary downstream metric; accuracy is omitted from the primary statistical summary because the final verified secondary analysis was unavailable.

Figure 12 summarizes the downstream Macro-F1 levels and the paired differences.

Table 8: Same-feature baseline specification and feature access in Final Phase A.

Policy	Score definition	Feature access
AdaLabel	$P(WC \mid s_{cheap}) - P(CW \mid s_{cheap})$	Full eligible cheap-side vector
Direct Linear	Linear predictor of ΔQ	Full eligible cheap-side vector
Direct MLP	MLP predictor of ΔQ	Full eligible cheap-side vector
WC-only	Estimated rescue component	Defining scalar score
Cheap-error	Estimated cheap-error likelihood	Defining scalar score
Entropy	Normalized cheap-label entropy	Defining scalar score
Margin	Top-1/top-2 cheap-label margin	Defining scalar score
Random	Uniform random ranking	No feature access
Oracle	Realized WC before CW	Non-deployable outcome access

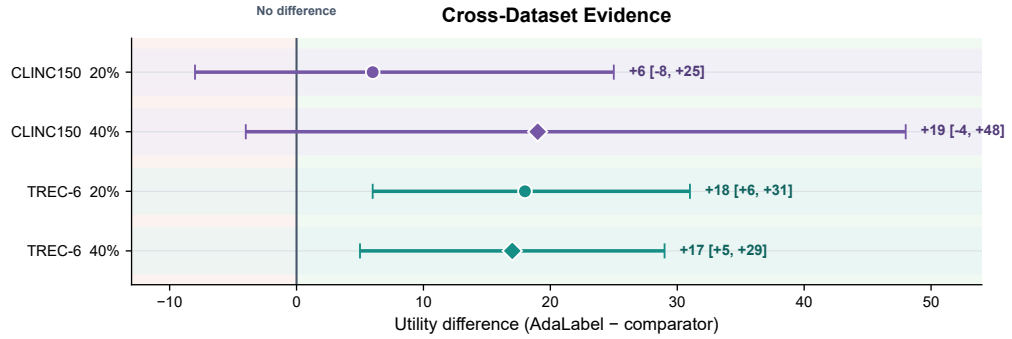

Figure 9: Parser-aware held-out Gold evidence on TREC-6 and CLINC150 at the frozen 20% and 40% budgets, visualizing Table 3. Points are observed utility differences and whiskers are reported paired bootstrap 95% intervals; the vertical line marks no difference. DBPedia14 is intentionally not pooled because its parser-free Gold interface and statistical evidence differ. TREC-6 intervals exclude zero, whereas CLINC150 intervals cross zero.

G.1 Seed-level paired Macro-F1

Table 11 reports per-seed Macro-F1 for both methods and the paired difference at each budget. Positive differences favor AdaLabel.

The parametric, bootstrap, and non-parametric robustness evidence for this comparison is consolidated in Appendix H.

Figure 13 plots the per-seed differences from Table 11.

G.2 Summary

AdaLabel achieves positive mean Macro-F1 gains at all three budgets. The improvement is statistically supported at the 40% budget under both parametric Holm correction and hierarchical bootstrap. The 20% and 30% differences are positive on average but their intervals include zero, indicating insufficient evidence rather than no effect. Annotation and training conditions are coupled, and the results should not be generalized to every downstream learner or task.

H Statistical Robustness

This section consolidates the paired, bootstrap, and non-parametric evidence for the Banking77 downstream comparison of AdaLabel versus the Direct MLP comparator.

AdaLabel shows a positive mean Macro-F1 gain at every budget, but only the 40% point is statistically supported: its Holm-adjusted p is 0.019 and its hierarchical-bootstrap 95% interval $[+0.00219, +0.02037]$ excludes zero. At 20% and 30% the mean differences are positive yet both the

Table 9: Annotator-hierarchy characterization on the frozen held-out TREC-6 evaluation population ($N = 500$). WC denotes rescue (cheap wrong, strong correct) and CW denotes regression (cheap correct, strong wrong); WC−CW is the net signed value. H1/H2/H3 are the three pre-specified Qwen3 cheap→strong hierarchies.

Hierarchy (cheap→strong)	Cheap Acc.	Strong Acc.	CC	WC	WW	CW	WC−CW
H1: 0.6B→1.7B	17.4	59.4	62	235	178	25	210
H2: 0.6B→8B	17.4	63.8	60	259	154	27	232
H3: 1.7B→8B	59.4	63.8	229	90	113	68	22

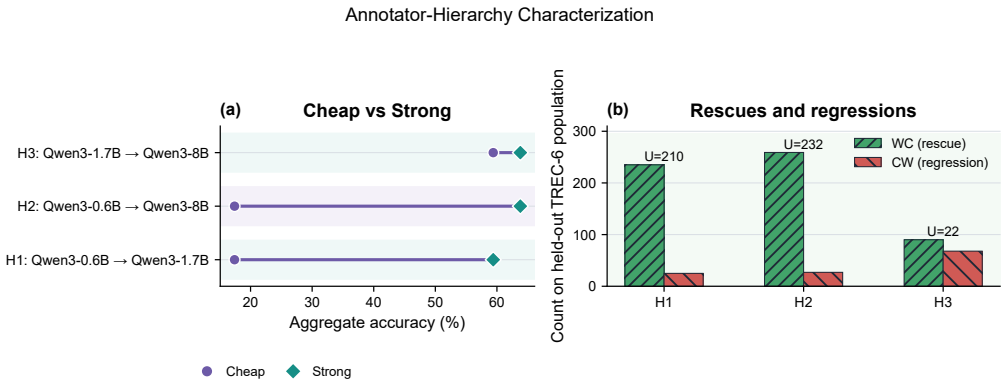

Figure 10: Annotator-hierarchy characterization on the frozen held-out TREC-6 population, visualizing Table 9. Panel (a) compares cheap and strong aggregate accuracies within each specified hierarchy; panel (b) separates WC rescues from CW regressions and annotates the frozen net value. In H3 the aggregate accuracy gap is small while both rescues and regressions remain substantial, so aggregate quality does not determine instance-level upgrade value.

paired and bootstrap intervals include zero (Holm $p = 0.200$ at both), so the evidence is insufficient rather than contradictory. The non-parametric Wilcoxon signed-rank test and the exact-binomial win-count test agree with this pattern (seed wins 10/15, 9/15, 12/15; exact-binomial p 0.30, 0.61, 0.035). A seed-only bootstrap (resampling seeds only) gives tighter intervals—[−0.00038, +0.01355] at 20% ($p = 0.065$), [−0.00101, +0.00839] at 30% ($p = 0.131$), [+0.00432, +0.01835] at 40% ($p = 0.003$)—because it ignores class-level variability; the hierarchical interval is preferred for reporting.

I Supporting Pre-Gold Evidence

The Phase A audit above uses the held-out Gold population. The same allocation logic is also evaluated on the pre-Gold Pool (11,984 DBPedia14 examples) as a supporting sanity check; this does not constitute an independent replication of the Gold audit.

The Pool results are broadly consistent with Gold at the larger budgets, where AdaLabel leads. Direct Linear leads Pool at 10%, and Cheap-error leads at 20%. These results provide supporting evidence, not an independent replication or a primary confirmatory claim.

Figure 14 plots the Pool budget curves behind Table 13.

Table 10: Matched-budget net utility $U = N_{WC} - N_{CW}$ across the three TREC-6 annotator hierarchies. H1=0.6B→1.7B, H2=0.6B→8B, H3=1.7B→8B. The bold entry in each hierarchy–budget column is the highest *deployable* utility; Oracle and Random are excluded from bolding. Random reports the mean over 1,000 rankings (SD 5.0–6.7). Oracle is a descriptive, non-deployable upper bound.

Hierarchy	Method	U@20%	U@30%	U@40%
H1	AdaLabel	**74**	101	117
H1	Direct MLP	59	92	122
H1	Direct Linear	68	**103**	122
H1	Cheap-error	67	93	**126**
H1	Entropy	51	69	79
H1	Margin	16	33	51
H1	WC-only	69	96	112
H1	Random (mean)	41.7	62.7	83.7
H1	Oracle (descriptive)	100	150	200
H2	AdaLabel	**66**	**95**	**124**
H2	Direct MLP	60	91	116
H2	Direct Linear	64	94	**124**
H2	Cheap-error	58	89	116
H2	Entropy	41	71	90
H2	Margin	38	60	76
H2	WC-only	63	94	118
H2	Random (mean)	46.1	69.4	92.7
H2	Oracle (descriptive)	100	150	200
H3	AdaLabel	23	23	27
H3	Direct MLP	**24**	33	**35**
H3	Direct Linear	18	22	27
H3	Cheap-error	17	27	28
H3	Entropy	21	**34**	26
H3	Margin	**24**	27	23
H3	WC-only	21	20	23
H3	Random (mean)	4.4	6.7	9.0
H3	Oracle (descriptive)	90	90	90

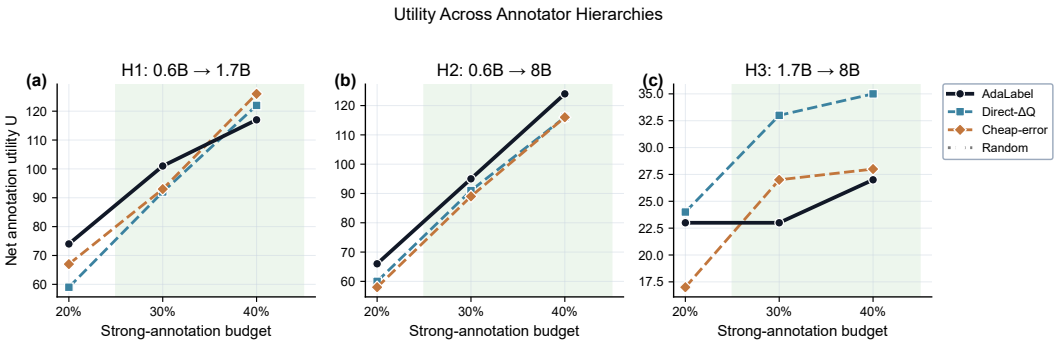

Figure 11: Matched-budget net utility on the frozen held-out TREC-6 population across the three specified annotator hierarchies, showing selected policies from Table 10. Direct-ΔQ denotes Direct MLP. Straight segments connect observed budgets only. AdaLabel is competitive and consistently above Random in these panels, but it is not uniformly dominant: several cells are led by other deployable methods. Panel-specific y-limits are retained to preserve resolution for H3.

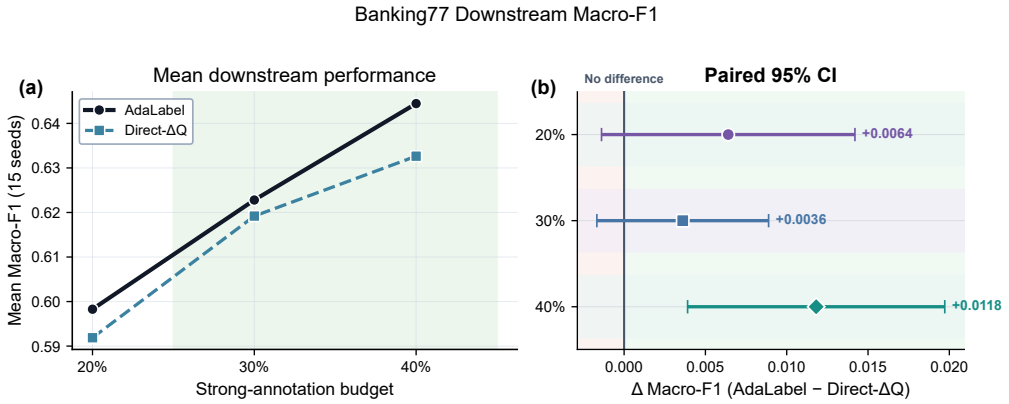

Figure 12: Banking77 downstream Macro-F1 under matched annotation budgets, averaged over the 15 frozen training seeds listed above. Direct-ΔQ denotes Direct MLP. Panel (a) shows method means without method-level intervals because the reported inferential intervals are paired differences; panel (b) shows AdaLabel minus Direct MLP paired mean differences and paired-t 95% CIs, where only the 40% interval excludes zero (see also Table 12). These results are restricted to the tested downstream protocol and do not imply a linear relation between direct utility and Macro-F1.

Table 11: Seed-level paired Macro-F1 on Banking77. Direct denotes Direct MLP; Diff = AdaLabel − Direct MLP. Scores and paired differences are rounded separately.

	20%			30%			40%		
Seed	AdaLabel	Direct	Diff	AdaLabel	Direct	Diff	AdaLabel	Direct	Diff
20260910	0.583	0.582	+0.001	0.616	0.620	−0.004	0.636	0.639	−0.003
20260911	0.608	0.584	+0.024	0.610	0.618	−0.008	0.627	0.631	−0.004
20260912	0.598	0.601	−0.003	0.619	0.608	+0.011	0.641	0.628	+0.013
20260913	0.614	0.594	+0.019	0.636	0.618	+0.018	0.634	0.633	+0.001
20260914	0.603	0.596	+0.007	0.623	0.625	−0.002	0.652	0.641	+0.011
20261101	0.594	0.604	−0.010	0.620	0.622	−0.001	0.656	0.633	+0.023
20261102	0.594	0.599	−0.004	0.625	0.624	+0.001	0.655	0.632	+0.022
20261103	0.565	0.559	+0.005	0.618	0.627	−0.009	0.616	0.638	−0.022
20261104	0.589	0.606	−0.016	0.627	0.622	+0.006	0.651	0.627	+0.024
20261105	0.605	0.611	−0.006	0.627	0.615	+0.012	0.646	0.631	+0.015
20261106	0.602	0.596	+0.007	0.624	0.622	+0.001	0.645	0.643	+0.003
20261107	0.613	0.592	+0.022	0.635	0.614	+0.021	0.660	0.631	+0.029
20261108	0.606	0.570	+0.036	0.625	0.623	+0.002	0.651	0.635	+0.017
20261109	0.601	0.594	+0.007	0.618	0.626	−0.007	0.650	0.626	+0.024
20261110	0.598	0.590	+0.007	0.618	0.605	+0.013	0.645	0.622	+0.024

Table 12: Consolidated downstream statistical robustness for AdaLabel versus the Direct MLP comparator on Banking77 (15 seeds, primary metric Macro-F1; differences are AdaLabel − Direct MLP). The 40% budget is the only one reaching significance after Holm-Bonferroni correction and excluding zero under the hierarchical bootstrap; the 20% and 30% intervals include zero. Accuracy is omitted from the primary summary because the final verified secondary analysis was unavailable.

Budget	Mean ΔF1	Paired 95% CI	Holm p	Cohen d_z	Seed wins	Wilcoxon p	Exact-bin p	Hier. boot 95% CI	Hier. boot p
20%	+0.0064	[−0.0014, +0.0142]	0.200	+0.46	10/15	0.107	0.30	[−0.00170, +0.01519]	0.129
30%	+0.0036	[−0.0017, +0.0089]	0.200	+0.38	9/15	0.252	0.61	[−0.00279, +0.01036]	0.260
40%	+0.0118	[+0.0039, +0.0197]	**0.019**	+0.83	12/15	**0.010**	**0.035**	[+0.00219, +0.02037]	**0.015**

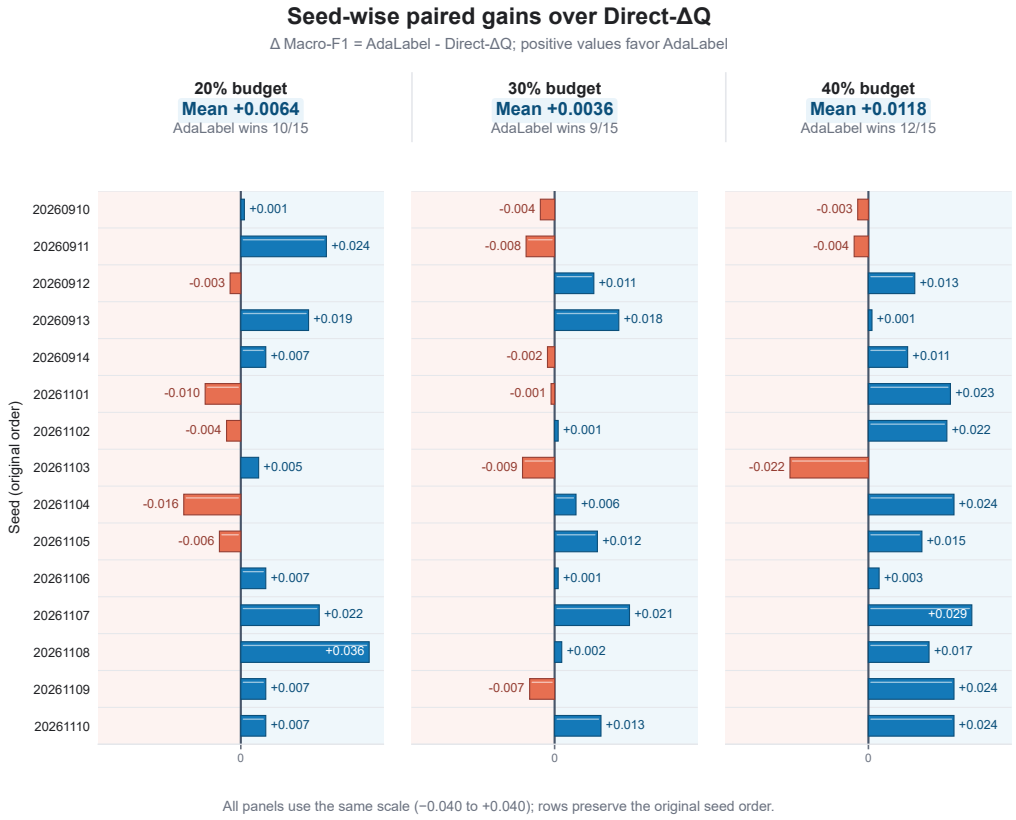

Figure 13: Seed-wise paired Banking77 Macro-F1 gains (AdaLabel minus Direct MLP) across the 15 frozen training seeds. Direct-ΔQ denotes Direct MLP. The figure visualizes the Diff columns of Table 11. Each cell uses the same signed range (-0.040 to $+0.040$); bars extending right favor AdaLabel and bars extending left favor Direct MLP. Rows preserve the original seed order. The compact summary reports the mean gain and the number of positive paired differences at each budget, while retaining the seed-level exceptions.

Table 13: Pre-Gold supporting sanity evidence on the DBPedia14 Pool population (selected counts 1,198 / 2,397 / 3,595 / 4,794 at 10%/20%/30%/40%). Net utility $U = N_{WC} - N_{CW}$. This evidence is supporting rather than the primary Gold evaluation.

Method	Pool 10%	Pool 20%	Pool 30%	Pool 40%
AdaLabel	568	965	1389	1786
Direct Linear	572	939	1259	1560
Direct MLP	421	796	1071	1258
WC-only	460	837	1244	1655
Cheap-error	564	977	1369	1751
Entropy	453	817	1089	1449
Margin	441	844	1244	1594

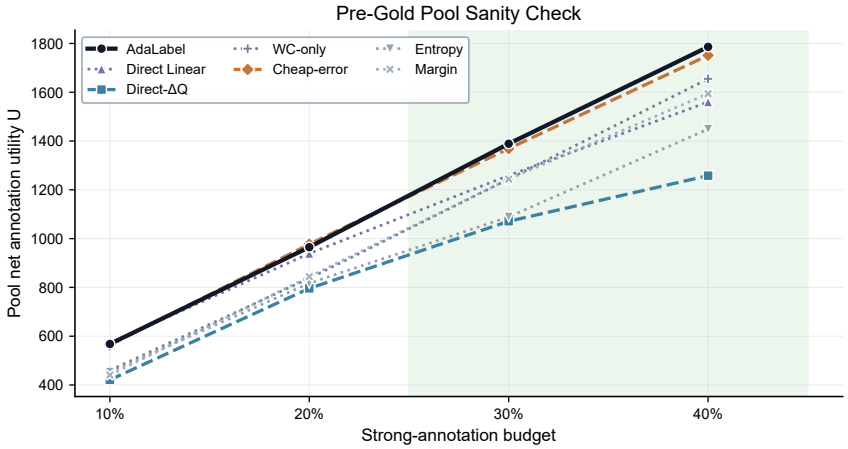

Figure 14: Supporting pre-Gold DBPedia14 Pool sanity check, visualizing the Pool columns of Table 13. Direct-ΔQ denotes Direct MLP. The plot shows matched-budget pool net utility for the same allocation policies; straight segments connect observed budget points only. This evidence is included as a sanity check and is not an independent confirmatory evaluation or a replacement for the held-out Gold audit.

J Protocol Clarifications and Validity Checks

This appendix records the protocol clarifications needed to interpret the main results and the final DBPedia14 same-feature audit. The aim is to make the information boundary explicit: AdaLabel estimates the signed value of an annotation upgrade from paired Meta outcomes, whereas target rankings are constructed from cheap-side state.

Table 14: Final protocol specification and information-access fairness. The original Gold protocol and independent prospective audits fix rankings before target strong outcomes are opened; the DBPedia14 all-method table is a corrective post-Gold same-feature audit.

Component	Final specification
Meta split	3,990 paired cheap/strong outcomes (DBPedia14 Meta)
Meta fold scheme	KFold(5, shuffle, seed=20261001), frozen
Meta V_{class}	cross-fitted from frozen KFold assignments (each fold uses the other four)
Target V_{class} (Pool/Gold)	computed from the full 3,990-row Meta outcome table
Scaler fitting population	fit on corrective Meta OOF feature matrix; applied unchanged to Meta/Pool/Gold
AdaLabel feature access	full eligible cheap-side vector
Direct Linear feature access	same full eligible cheap-side vector
Direct MLP feature access	same full eligible cheap-side vector
Gold strong-outcome access at ranking	none in the original prospective Gold protocol and independent prospective audits; final DBPedia14 all-method table is corrective post-Gold
Budget definition	fixed strong-call counts: 402 / 804 / 1,206 / 1,608 (10%/20%/30%/40%)
Budget prefixes	nested prefixes of one decision rule (same ranking supplies every budget)
Tie-breaking	sample_id ascending
Annotator hierarchy	cheap Qwen3-1.7B, strong Qwen3-8B; both frozen, temperature 0, seed 20260830

Q1. What information is available when a target ranking is constructed? The ranking uses only cheap-side state available before an upgrade. This state may include the cheap annotator's class-score distribution, derived uncertainty summaries, the cheap-predicted class, and Meta-derived statistics such as V_{class}. It cannot include the strong annotation $a_s(x)$, a strong-side score, the realized CC/WC/WW/CW outcome, or any post-upgrade label for the target example.

Q2. Are learned methods given the same input representation? Yes. AdaLabel, Direct Linear, and Direct MLP use the same full eligible cheap-side feature vector. The scalar policies retain their defining scalar scores: WC-only uses the estimated rescue component, Cheap-error uses estimated cheap-error likelihood, Entropy uses normalized cheap-label entropy, and Margin uses the top-1/top-2 cheap-label margin. Oracle is non-deployable because it uses realized outcomes.

Q3. How is V_{class} constructed without target leakage? For cheap-predicted class k, $V_{\text{class}}(k) = (N_{WC,k} - N_{CW,k})/(N_k + 1)$. Meta rows use out-of-fold construction: each validation fold receives a value computed from the other four frozen KFold folds. Pool and Gold rows use the full 3,990-row Meta outcome table. No Pool or Gold strong outcome enters this statistic.

Q4. Is the final DBPedia14 all-method table prospective? No. The final DBPedia14 all-method table is a corrective post-Gold same-feature audit on the frozen Gold population. The original Gold protocol fixed rankings before Gold strong outcomes were accessed. The corrective audit is separate from that prospective claim and preserves the Gold population, outcomes, budgets, ranking rule, estimator configurations, and baseline set.

Q5. Where is independent prospective confirmation provided? The independent prospective confirmation is the Yahoo Answers resource-aware audit. It uses disjoint Meta and Gold splits

of 2,000 and 4,000 examples, respectively, with zero overlap. The feature schema, Meta training configuration, rankings, and nested selected sets were sealed before Gold strong access. Methods were compared under matched logical-token budgets, and replay was checked with CUDA-event timing.

Q6. Why is the AdaLabel estimator intentionally lightweight? The estimator is deliberately lightweight because the scientific question concerns signed annotation value rather than predictive architecture capacity. AdaLabel estimates a four-outcome distribution and then derives $P(WC \mid s_{cheap}) - P(CW \mid s_{cheap})$. More expressive predictors could be substituted if they estimate the same conditional outcome probabilities.

Q7. Why model four states instead of directly predicting ΔQ? The scalar utility depends on the contrast between WC and CW, but the four-state model preserves the full outcome distribution. CC and WW both have zero signed utility, yet they differ scientifically: CC is joint success, and WW is joint failure. The Amazon replication supports predictive separability of these zero-utility states (AUROC 0.8099, or 0.810 rounded). It does not establish universal calibration superiority or signed-utility dominance.

Q8. Does AdaLabel always beat every deployable baseline? No. On DBPedia14 Gold, AdaLabel leads deployable methods at 10%, 30%, and 40%, while Cheap-error leads at 20% by 13 utility units. In the TREC-6 hierarchy study, other deployable methods lead several hierarchy-budget cells. On Yahoo Answers, WC-only matches or exceeds AdaLabel at some budgets, although AdaLabel exceeds Direct Linear at every tested budget. The claim is signed allocation evidence under matched budgets, not universal dominance.

Q9. How are different budgets generated? Each policy produces one complete ranking for the relevant target population. The 10%, 20%, 30%, and 40% rows are nested prefixes of that ranking. This rule prevents outcome-driven budget search within a policy and makes budget curves comparable across methods.

Q10. How are ties handled? Ties are broken by ascending `sample_id`. This rule is part of the ranking specification and is applied consistently across policies that can produce equal scores.

Q11. What exactly is the unit of budget? DBPedia14 and Banking77 use matched strong-call counts. On DBPedia14 Gold, the 10%, 20%, 30%, and 40% budgets correspond to 402, 804, 1,206, and 1,608 upgrades. Banking77 upgrades exactly 1,518, 2,277, and 3,036 of the same 7,588 eligible training examples at 20%, 30%, and 40%. Yahoo Answers instead matches logical-token budgets and also reports CUDA-event GPU time. These accounting units are not equivalent to universal commercial API cost, latency, or monetary price.

Q12. Does higher annotation utility guarantee downstream improvement? No. Net annotation utility measures direct correctness change in selected labels, $U = N_{WC} - N_{CW}$. Downstream performance depends on the learner, optimizer, data distribution, and the interaction between label changes and training. Banking77 is therefore reported as a separate downstream experiment: AdaLabel has positive mean Macro-F1 differences at all three tested budgets, while only the 40% budget is statistically supported after Holm correction and hierarchical bootstrap.

Q13. What assumptions are required for AdaLabel to transfer? AdaLabel requires representative paired Meta supervision and a stable cheap/strong hierarchy. Transfer can fail if the allocation population differs in class composition, language, difficulty, annotation interface, or annotator behavior. These assumptions are part of the method's applicability boundary and are tested through held-out and cross-dataset evidence rather than assumed away.

Q14. What is the exact novelty claim? The contribution is the formulation and prospective estimation of signed annotation value for a non-oracle stronger annotator. AdaLabel values the intervention of replacing an existing cheap annotation with a stronger annotation, using expected rescue minus expected regression. We do not claim to invent the CC/WC/WW/CW symbols themselves, and we do not claim that AdaLabel is universally dominant.

K REPRODUCIBILITY

Code availability. Anonymous code and reproducibility materials are provided in the supplement accompanying this submission. Its top-level `README.md` maps each paper claim to the corresponding files. Phase-specific entry points are `experiments/phase_a/FINAL/README.md`, `experiments/phase_b/FINAL/README.md`, and `reproducibility/README.md`.

All numerical results reported in the paper are reproducible offline from the released frozen records or directly verifiable from the released paper-facing source tables. DBPedia14 provides frozen feature/source tables, scores, rankings, result tables, and an exact verification program; Banking77 provides the aggregate paper-facing source table, while seed-level runs and checkpoints for these two legacy experiments are intentionally excluded. CLINC150, TREC-6 P9, and Yahoo Answers provide frozen sample-level records and standalone verification or reproduction programs. The Amazon and exploratory GoEmotions four-state studies provide frozen score/source tables and statistical reproduction materials; the signed-value ablation and centered CW-heterogeneity analyses are reproducible from released Yahoo Answers and 20 Newsgroups records. The 20 Newsgroups clean-room study provides its protocol, records, predictions, features, models, rankings, results, and end-to-end scripts, while the separate TREC-6 G3 package provides frozen predictions, features, result tables, checksums, and an audit program. Public base-model weights are not redistributed.

The scientific unit of comparison is an example-level annotation upgrade. Sample identifiers are preserved across rankings, call-budget comparisons use equal selected counts, and utility is computed from shared outcome labels. Yahoo Answers instead matches logical-token budgets. Meta supervision is separated from Pool and Gold evaluation. In the original Gold protocol, rankings are fixed before Gold strong outcomes are accessed. The final same-feature table is labeled separately as a corrective post-Gold audit rather than as a new prospective ranking-freeze claim.

The supplement includes the DBPedia14 feature/source tables, rankings, scores, outcome tables, and matched-budget result tables; these can be checked with `scripts/verification/verify_phase_a.py`. Banking77 is represented by the aggregate source table used in the paper. File checksums are listed in `reproducibility/SHA256SUMS.txt`. No new data access or inference is required to verify these reported table values.

L EXTENDED LIMITATIONS

AdaLabel is supported for a particular upgrade setting rather than every strong-model deployment. First, it requires paired Meta supervision containing cheap and strong annotations plus reference labels. Collecting these outcomes consumes annotation compute, and the method is less attractive when representative paired data cannot be obtained. Rescue and regression probabilities cannot be learned from cheap-side features alone without this supervision.

Second, transfer from Meta can fail under distribution shift. Differences in class composition, language, difficulty, annotation interface, or annotator behavior may miscalibrate the conditional outcome probabilities. The DBPedia14 Gold study is held out, but the final same-feature table is a corrective post-Gold audit rather than a new prospective claim. A new task would require checking both the Meta representation and the stability of the cheap/strong hierarchy.

Third, results depend on the annotator pair. A different model family, prompt, decoding protocol, class verbalizer, or reference standard can change which examples are WC and CW and therefore which ranking is useful. The strong annotator is not assumed to be an oracle. A hierarchy with negligible improvement or uniformly positive upgrades would pose a different allocation problem and may offer less benefit from a signed model.

Fourth, the cost model remains incomplete. The original DBPedia14 and Banking77 experiments match strong-call counts, not monetary price, token use, latency, or batching; Yahoo additionally matches logical-token budgets and measures CUDA-event time. Future work could extend signed allocation to annotator-specific values and heterogeneous costs, jointly selecting examples and annotation sources under a shared budget.

Fifth, cross-dataset evidence is heterogeneous. TREC-6 intervals exclude zero in its parser-aware test, CLINC150 intervals include zero, and DBPedia14 provides a corrective post-Gold audit with a

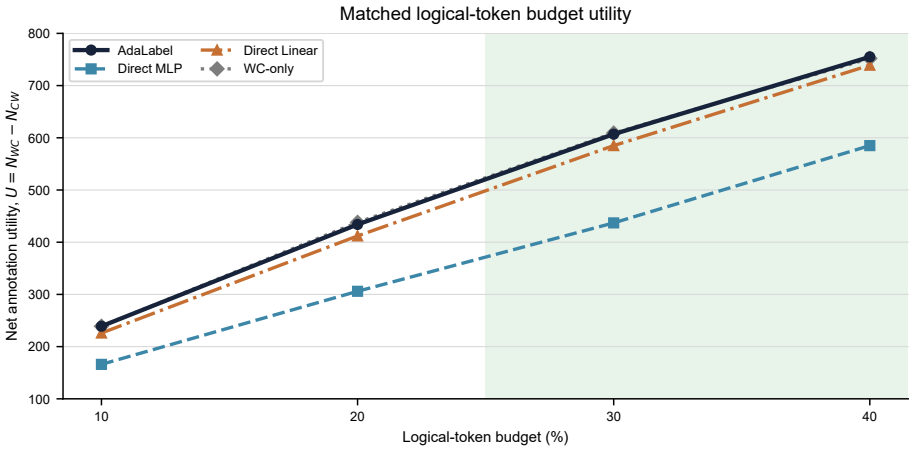

Figure 15: Yahoo Answers allocation utility under matched logical-token budgets. All curves use frozen nested ranking prefixes on the same 4,000-example Gold split. Colors, line styles, and markers distinguish the four methods, with AdaLabel emphasized in dark blue. The light-green background highlights the 30–40% operating range and is not a confidence interval. The figure reports allocation utility only.

parser-free interface. Interface validity and parser behavior can alter observed correctness. These records therefore delimit rather than prove transfer.

Sixth, the experiments concern classification-style annotation over a finite set of tasks and annotator models. Open-ended generation, multilabel or structured prediction, multimodal annotation, and settings without a clear correctness reference may require other outcome taxonomies or utility functions. The four-state correctness utility is useful for the evaluated setting but is not uniquely applicable to all annotation tasks.

Finally, downstream evidence remains bounded. The Banking77 experiment shows that improved annotation allocation translates into higher downstream RoBERTa Macro-F1 under matched budgets, but annotation and training budgets are coupled. The results provide empirical support for the allocation-to-performance connection under the tested conditions but do not establish that every learner or task benefits from a higher-utility allocation. Net annotation utility and downstream performance should therefore continue to be reported as related but distinct quantities.

M Yahoo Answers Resource-Aware Evaluation

The independent prospective Yahoo Answers evaluation used disjoint Meta and Gold splits of 2,000 and 4,000 examples, respectively (overlap zero). All ten candidate class labels were evaluated using the frozen teacher-forced scoring procedure. The cheap and strong annotators were Qwen3-1.7B and Qwen3-8B at BF16. Candidate microbatch sizes were two for cheap scoring and one for strong scoring. The strong microbatch size was amended to avoid out-of-memory errors before Gold strong outcomes were opened. The tokenizer, template, candidate order, scoring semantics, and model revision were unchanged. The feature schema, Meta training configuration, rankings, and nested selected sets were sealed before Gold strong access.

Logical token accounting included candidate-scoring work for all candidate labels. At each 10%, 20%, 30%, and 40% budget, a method selected the longest frozen ranking prefix that did not exceed its logical-token budget. Utility was $U = N_{WC} - N_{CW}$. Across the 4,000 Gold examples, the outcome counts were CC=474, WC=1,472, WW=1,774, and CW=280; cheap and strong accuracy were 0.1885 and 0.4865, and both the accuracy difference and $(WC - CW)/4{,}000$ were 0.2980.

Figure 15 shows the utility curves. The primary comparison was AdaLabel against Direct MLP. It used 10,000 reference-class-stratified bootstrap replicates with seed 20260921, a two-sided plus-one

Table 15: Exploratory frozen-record comparison between SAV and WC-only on Yahoo Answers. Rankings are deterministically replayed from the sealed estimator and exactly reproduce the released frozen orderings. Ada-only and WC-only-only denote examples selected exclusively by the corresponding policy under the matched logical-token budget. Predicted regression risk is the frozen estimator's $P(CW \mid s_{cheap})$; realized CW rate uses frozen Gold outcomes. Δ swap utility is $U(\text{Ada-only}) - U(\text{WC-only-only})$. This diagnostic is descriptive and is not a new confirmatory experiment.

Budget	Ada n	WC n	Jaccard	Ada-only n	WC-only-only n	Ada-only $\overline{P(CW)}$	WC-only-only $\overline{P(CW)}$	Ada-only CW rate	WC-only-only CW rate	Ada-only U	WC-only-only U	Δ swap U
10%	459	459	0.796	52	52	0.0258	0.0977	0.000	0.019	29	29	0
20%	914	908	0.827	89	83	0.0344	0.1077	0.011	0.048	38	42	-4
30%	1356	1380	0.829	116	140	0.0369	0.1108	0.017	0.086	50	52	-2
40%	1776	1788	0.835	154	166	0.0418	0.1169	0.032	0.127	51	48	+3

p-value, and Holm correction over four budgets. The adjusted p-values in the corrected statistical records are all approximately 0.0008 and are reported as $p < 0.001$.

The replay archive contains 24 successful single-GPU runs: AdaLabel and Direct MLP each have four nested logical-token increments with three repetitions. The runner records GPU duration with separate CUDA events bracketed by synchronization and asserts replay top-1 agreement with the sealed Gold strong label. At 40%, cumulative values are the sum of the four increments within a repetition: AdaLabel has 2,763,196 logical tokens and 696.46 CUDA-event GPU seconds per complete replay; Direct MLP has 2,761,116 tokens and 672.07 seconds. The corresponding aggregate throughputs are 3,967 and 4,108 logical tokens per CUDA-event GPU second.

This prospective evaluation is limited to the tested annotator/interface pair, for which absolute Yahoo accuracies are low. WC-only matches or exceeds AdaLabel at some budgets. Token matching does not guarantee equal commercial-API price or latency.

M.1 Exploratory Yahoo SAV–WC-only Frozen-Record Diagnostic

This descriptive diagnostic replays the sealed Yahoo estimator on frozen Gold features. We loaded the frozen StandardScaler and four-state LogisticRegression, then applied only `transform` and `predict_proba`; no estimator or scaler was refit, no LLM inference was performed, and no ranking or Gold outcome was altered. The serialized `classes_` order is [CC, CW, WC, WW], which differs from the display-schema order. We therefore indexed probabilities by semantic class label rather than an assumed column position. The replay exactly reproduced both released 4,000-example rankings, with zero rank mismatches, no score ties, and exact selected prefixes at all four logical-token budgets.

SAV and WC-only rankings were highly correlated (Spearman $\rho = 0.9614$; Kendall $\tau = 0.8370$), but 8.15% of strict pairwise orderings differed. Across all budgets, Ada-only sets had lower mean predicted $P(CW \mid s_{cheap})$ and lower realized CW rates than WC-only-only sets. However, the swap utility was not uniformly positive because WC-only-only sets could also contain more WC rescues. Thus, on Yahoo, the CW term acts as a targeted regression-risk correction rather than a universally beneficial ranking perturbation. This frozen-record analysis is exploratory and does not alter the primary prospective result.

N Controlled Regression-Risk Heterogeneity Replay

This exploratory frozen-record mechanism analysis generalizes the Yahoo endpoint diagnostic in Appendix M.1 to Yahoo Answers and 20 Newsgroups. It made no new annotator calls and did not refit any estimator, scaler, prior, or feature transformation. For each dataset, let $\bar{p}_{CW} = N^{-1} \sum_i p_i(CW)$ and define

$$p_{i,\lambda}(CW) = \bar{p}_{CW} + \lambda\big[p_i(CW) - \bar{p}_{CW}\big], \qquad V_\lambda(x_i) = p_i(WC) - p_{i,\lambda}(CW),$$

on the fixed grid $\lambda \in \{0, 0.1, \ldots, 1\}$. Centering separates heterogeneity from mean risk because

$$\mathbb{E}[p_{CW,\lambda}] = \mathbb{E}[p_{CW}], \qquad \mathrm{Var}(p_{CW,\lambda}) = \lambda^2 \,\mathrm{Var}(p_{CW}).$$

Numerically, mean $p_{CW,\lambda}$ remained 0.072859 on Yahoo and 0.012333 on 20 Newsgroups at every grid point. The largest absolute variance-identity error was below 8×10^{-34}. Exact sample-identifier

Table 16: Endpoints of the centered frozen-record replay. Reversal and overlap are percentages. Differences are relative to WC-only; negative ΔN_{CW} denotes fewer realized regressions.

Dataset	Budget	λ	Reversal	Overlap	ΔN_{WC}	ΔN_{CW}	ΔU
Yahoo	10%	0	0.00	100.00	0	0	0
	20%	0	0.00	100.00	0	0	0
	30%	0	0.00	100.00	0	0	0
	40%	0	0.00	100.00	0	0	0
	10%	1	8.15	88.67	-1	-1	0
	20%	1	8.15	90.26	-7	-3	-4
	30%	1	8.15	91.45	-12	-10	-2
	40%	1	8.15	91.33	-13	-16	$+3$
20 Newsgroups	10%	0	0.00	100.00	0	0	0
	20%	0	0.00	100.00	0	0	0
	30%	0	0.00	100.00	0	0	0
	40%	0	0.00	100.00	0	0	0
	10%	1	1.56	99.01	$+2$	-2	$+4$
	20%	1	1.56	99.36	$+1$	-1	$+2$
	30%	1	1.56	99.24	$+5$	-3	$+8$
	40%	1	1.56	99.22	-3	-3	0

tie-breaking gave zero endpoint rank mismatches: $\lambda = 0$ reproduced WC-only, and $\lambda = 1$ reproduced each released SAV ranking. At $\lambda = 0$, Spearman ρ, Kendall τ, and selected-set overlap were exactly one.

Table 16 summarizes the two endpoints. Pairwise reversal is computed exactly from Kendall τ, not by sampled pairs. Selection overlap is $|S_\lambda \cap S_{WC}|/|S_\lambda|$. Across all 88 dataset–budget–λ conditions, integer bookkeeping verified

$$\Delta U = \Delta N_{WC} - \Delta N_{CW}.$$

Increasing heterogeneity generally made ΔN_{CW} more negative, but ΔU was non-monotonic because the changed selections could also lose WC rescues.

Swap sets and resource matching. For each grid point and budget, the replay records $S_\lambda \setminus S_{WC}$ and $S_{WC} \setminus S_\lambda$, including frozen mean $P(WC)$ and $P(CW)$, realized WC/CW rates, and swap utility. Yahoo retains its original logical-token budgets, so different ranking prefixes can contain different numbers of examples; its two exclusive sets therefore need not have equal size. The fixed call-count budgets on 20 Newsgroups produce equal-size exclusive sets exactly. The full Yahoo endpoint values remain in Table 15; complete grid-level records are supplied as CSV rather than duplicated in the PDF.

Predictability and uncertainty. Using realized CW as the positive class, frozen $P(CW)$ achieved AUROC 0.6754, AUPRC 0.1319, and Brier score 0.0636 on Yahoo. The corresponding 20 Newsgroups values were 0.6639, 0.0563, and 0.0164. These values indicate usable but imperfect discrimination and do not imply strong calibration. Ten-bin reliability records are included in the supplement. For ΔN_{WC}, ΔN_{CW}, and ΔU, nominal 95% percentile intervals use 10,000 paired bootstrap replicates stratified by reference class with fixed seed 20261201. These mechanism analyses are exploratory and are not multiplicity-adjusted claims of superiority.

O Isolated AdaLabel–Cheap-error A/B Battle on 20 Newsgroups

We report a supplementary descriptive head-to-head comparison on the fresh 20 Newsgroups Gold population ($N = 13{,}193$). Both policies use identical strong-annotation budgets. Utility is $U = N_{WC} - N_{CW}$, ΔU is AdaLabel minus Cheap-error, and the intervals are unadjusted paired-bootstrap 95% intervals. None of the four budget-level comparisons is significant after Holm correction.

Table 17 and Figure 17 show AdaLabel marginally ahead of Cheap-error at the observed budgets. Table 18 reports a separate, non-exhaustive set of contextual comparisons. This supplementary comparison remains descriptive. The anonymous clean-room supplement contains the frozen split

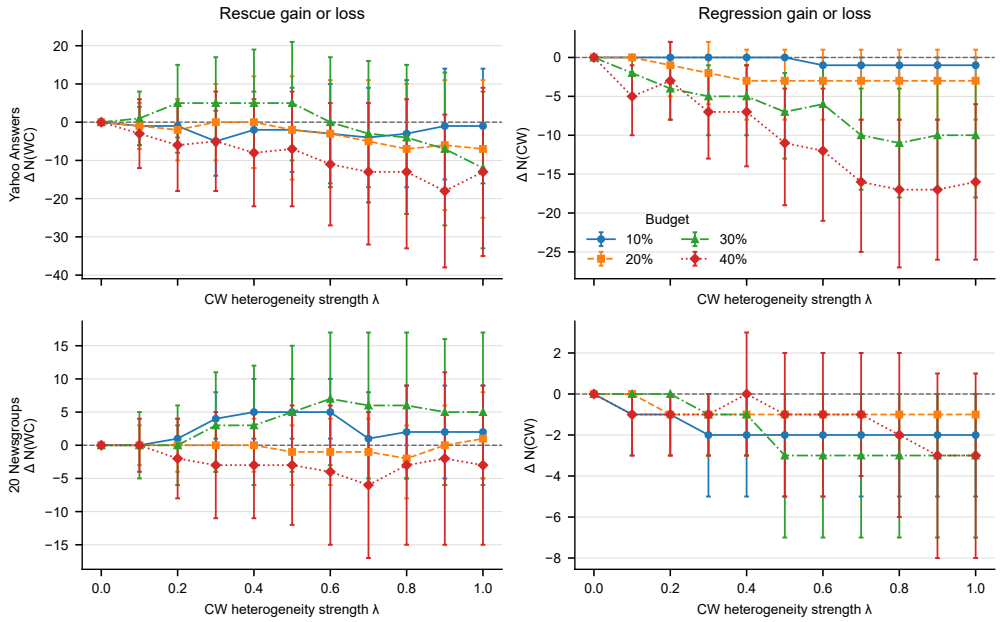

Figure 16: Rescue–regression decomposition for the exploratory centered frozen-record replay. The left column shows ΔN_{WC}, the change in realized rescues relative to WC-only; the right column shows ΔN_{CW}, the change in realized regressions. Yahoo Answers is shown above 20 Newsgroups. Together the two components determine signed utility through $\Delta U = \Delta N_{WC} - \Delta N_{CW}$. Error bars are nominal 95% reference-class-stratified paired-bootstrap intervals from 10,000 replicates. Negative and non-monotonic results are retained.

Table 17: AdaLabel–Cheap-error A/B comparison on fresh 20 Newsgroups Gold; AdaLabel is slightly higher at every observed budget.

Budget	AdaLabel U	Cheap-error U	ΔU	95% CI
10%	**800**	726	+74	[14, 134]
20%	**1484**	1393	+91	[13, 168]
30%	**2156**	2033	+123	[33, 213]
40%	**2761**	2679	+82	[−13, 177]

and protocol, model and feature artifacts, rankings, selected sets, outcome tables, paired-bootstrap records, and reproduction entry points needed to reproduce the reported 20 Newsgroups utilities, intervals, table, and battle figure.

P Signed-Value Attribution Ablation

The attribution analysis reuses the frozen Meta features, Gold cheap-side features, Gold outcomes, and budget rules from Yahoo Answers and 20 Newsgroups. It makes no new annotator calls and does not use DBPedia14 or Banking77. Within each dataset, all learned variants receive identical Meta supervision and cheap-side features. The three-state target maps CC and WW to ZERO while retaining WC and CW. The two-correctness variant fits separate binary correctness models for the cheap and strong annotators and subtracts their predicted probabilities. The class-only policy ranks by the Meta-derived mean signed value of the cheap-predicted class, using additive denominator smoothing by one. Yahoo uses the same 20-feature logistic configuration and matched logical-token prefixes as its frozen analysis. 20 Newsgroups uses the same standardized 66-feature input, $(64, 32)$ MLP architecture, seed, and matched call-count prefixes as its frozen four-state analysis.

Table 18: Contextual utilities for non-battle baselines on the same frozen 20 Newsgroups Gold population. This compact reference is separate from the AdaLabel–Cheap-error A/B comparison.

Method	10%	20%	30%	40%
AdaLabel	**800**	**1484**	**2156**	**2761**
Direct MLP	746	1422	2091	2698
WC-only	796	1482	2148	**2761**
Entropy	652	1286	1911	2522
Margin	603	1228	1866	2473

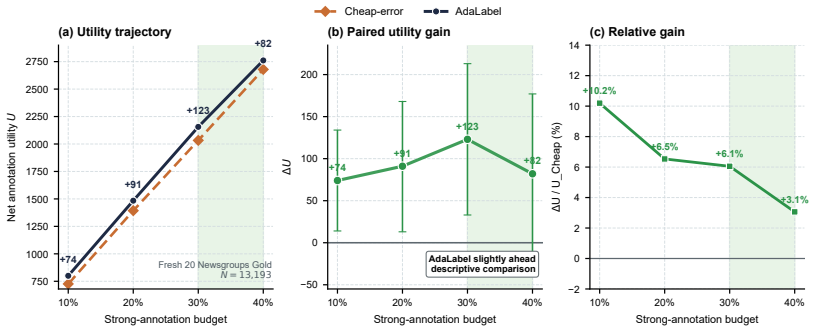

Figure 17: AdaLabel–Cheap-error A/B battle at matched strong-annotation budgets on fresh 20 Newsgroups Gold. AdaLabel has positive point-estimate differences at all four budgets.

The class-only comparisons provide exploratory evidence that the learned score contributes within-class information beyond class composition. The three-state comparison provides limited positive evidence for retaining CC and WW as distinct training outcomes, concentrated at particular budgets. The two-correctness results show that the signed target is not tied to a unique parameterization. They also bound the utility claim: the four-state estimator is competitive rather than uniformly superior to an alternative estimator of the same signed quantity. The released source table records WC and CW decompositions, selected counts, and actual Yahoo token expenditure for every row.

Q Four-State Outcome-Auditability Replication

The auditability study used disjoint Meta and Gold splits. GoEmotions was exploratory (2,000 Meta and 4,000 Gold examples): its 13 Meta CC examples made the CC–WW estimate too imprecise for confirmation. The prospectively frozen Amazon Reviews Multi English replication used 8,000 proportionally sampled training examples for Meta and 4,000 test examples for Gold (five star classes; seed 5090; revision `ec73b665e4be0f567b69d39425355401cfe0d29b`). Qwen3-0.6B was cheap, Qwen3-1.7B supplied Meta outcomes, and Qwen3-8B Gold outcomes; runtime, models, cheap-side probabilities, rankings, and selection identifiers were sealed before Gold strong scoring.

On Amazon, Meta-only five-fold out-of-fold predictions separated CC from WW (1,313 and 2,983 examples; AUROC 0.8099, 95% CI [0.7966, 0.8226], one-sided 10,000-permutation $p = 0.0001$; CC-positive AUPRC 0.6241). On Gold, four-state $P(WW)$ reached 0.5530 AUROC versus 0.5165 for three-state $P(\text{ZERO})$ (paired difference 0.0365, 95% CI [0.0137, 0.0589], $p = 0.0018$); the exploratory GoEmotions difference was 0.0066 (95% CI [0.0012, 0.0121]). The Amazon Brier difference was 0.00014 (95% CI $[-0.00032, 0.00059]$), supporting predictive separation and improved joint-failure detection, not general calibration or signed-utility superiority. Released scores, calibration analyses, checksums, and reproduction entry points are in the anonymous supplement.

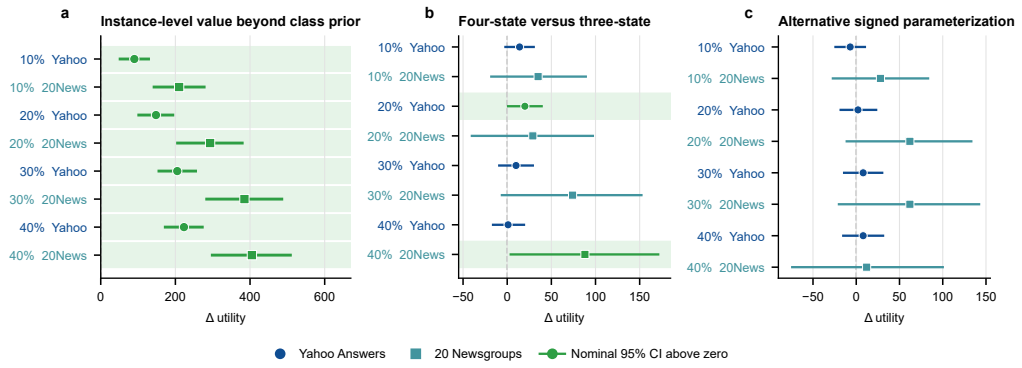

Figure 18: Attribution of signed-value allocation on frozen Yahoo Answers ($N = 4{,}000$ Gold) and 20 Newsgroups ($N = 13{,}193$ Gold). Points are four-state utility minus the stated comparator; whiskers are nominal 95% percentile intervals from 10,000 paired, reference-class-stratified bootstrap resamples. Circles denote Yahoo Answers and squares denote 20 Newsgroups. Green rows highlight intervals lying above zero before correction for the newly examined dataset–budget–comparator family. (a) The observed gaps consistently favor the full estimator over the class-only signed prior, providing exploratory evidence for instance-level value learning beyond class composition. (b) Separating CC and WW yields budget-dependent gains over the matched three-state model. (c) The four-state model remains competitive with the alternative two-correctness parameterization of the same signed decision quantity, but every interval includes zero. Source data are provided in the supplementary analysis package.

Table 19: Paired attribution effects. Entries are four-state minus comparator utility with nominal 95% percentile intervals from 10,000 reference-class-stratified paired bootstrap resamples. Bold entries have intervals above zero before correction for the newly examined family; these exploratory intervals are not multiplicity-adjusted.

Dataset	Comparator	10%	20%	30%	40%
Yahoo Answers	Three-state	+14 [−2, 30]	**+20 [1, 39]**	+10 [−9, 29]	+1 [−16, 19]
Yahoo Answers	Two-correctness	−7 [−24, 10]	+2 [−18, 23]	+8 [−14, 30]	+8 [−15, 31]
Yahoo Answers	Class-only prior	**+90 [52, 128]**	**+148 [102, 193]**	**+205 [156, 254]**	**+223 [173, 272]**
20 Newsgroups	Three-state	+35 [−18, 89]	+29 [−40, 97]	+74 [−6, 152]	**+88 [4, 171]**
20 Newsgroups	Two-correctness	+28 [−27, 83]	+62 [−11, 133]	+62 [−20, 142]	+12 [−74, 100]
20 Newsgroups	Class-only prior	**+210 [143, 277]**	**+293 [206, 379]**	**+385 [284, 485]**	**+405 [299, 508]**