CALE: COST-AWARE LARGE–SMALL ENSEMBLE FOR HETEROGENEOUS LLM ANNOTATION POOLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly used as annotators, and organizations often have access to heterogeneous pools of large and small LLMs with different costs and capabilities. We study the practical yet underexplored setting of annotating a fixed pool of unlabeled instances under a finite budget, where both labels and model reliability are initially unknown. The goal is to allocate this budget across models and instances to obtain high-quality annotations. We propose CALE, a cost-aware large–small ensemble framework that couples label aggregation with sequential acquisition. CALE models heterogeneous annotator reliability hierarchically, sharing information across similar instances while preserving instance-specific variation, and treats repeated calls to the same model as dependent rather than independent evidence. It then selects each model–instance query using complementary gains from improving the current label and learning reliability for future decisions, normalized by query cost. The resulting closed-loop interaction allows aggregation and acquisition to reinforce each other throughout the annotation process. Across five benchmarks spanning topic classification, clinical inference, entailment, ontology classification, and multi-subject question answering, annotated by heterogeneous LLM pools, CALE achieves a higher area under the accuracy–cost curve (AUBC) than every other baseline on every dataset, with the largest gains at tight budgets: it reaches 85.5% mean final accuracy at $1.5N$, whereas competing systems require about $2.9N$ to reach the same level. Ablation and robustness studies analyze the sources of the gains and assess stability across modeling and deployment choices.

1 INTRODUCTION

Large language models (LLMs) are reshaping how organizations label data at scale, largely reducing the need for manual annotation (Tan et al., 2024; Ding et al., 2023; He et al., 2024; Farr et al., 2024). In practice, an organization often has access to a heterogeneous pool of large and small LLMs whose costs and capabilities differ substantially (Wang et al., 2021; Chen et al., 2023; Huang et al., 2025). We consider a practical setting in which an organization seeks to label a fixed pool of unlabeled data instances under a finite budget using a heterogeneous pool of strong-but-expensive and cheap-but-weaker models. The central question is how to allocate this budget across models and instances to maximize annotation quality, without access to gold labels during the algorithm's execution and with unknown model reliability. The goal is therefore to maximize annotation accuracy throughout the cost trajectory, rather than only at the terminal budget.

Under this setting, this paper studies a different object from two nearby lines of research: LLM routing and cascading, and crowdsourcing aggregation. *First*, LLM routing and cascading methods typically seek a favorable quality–cost trade-off (Shnitzer et al., 2023; Ding et al., 2024; Zhang et al., 2024; Dekoninck et al., 2024). Routing decides whether each incoming request should use a strong-but-expensive or a cheap-but-weaker model, whereas cascading follows a sequential strategy: it first relies on a cheaper model and turns to a stronger one only when the initial answer appears unreliable. Instead of routing or cascading for each individual instance, our setting focuses on allocating a global budget over a fixed unlabeled pool. *Second*, crowdsourcing aggregation methods focus on inferring reliable labels from multiple noisy annotators, often by jointly estimating latent labels and annotator reliability (Dawid & Skene, 1979; Whitehill et al., 2009; Raykar et al., 2010; Li et al., 2019a;b). However, existing routing, cascading, and selection methods commonly rely on

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.

model-provided uncertainty or confidence scores, or on labeled historical data, to decide when and where to route (Dekoninck et al., 2024; Farr et al., 2024; Ong et al., 2025). Other approaches train an auxiliary router, selector, or judge model, which introduces maintenance overhead when the data distribution or model pool changes (Chen et al., 2023; Ding et al., 2024; Ong et al., 2025; Maurya et al., 2025). These requirements are not available or desirable in our setting, where the algorithm must learn from its acquired responses without auxiliary training.

To bridge this research gap, we propose CALE (Cost-Aware Large–small Ensemble), which addresses the shared needs of organizations and unites the complementary strengths of both research streams. CALE realizes this coupled process through two modules: (a) a hierarchical aggregation module and (b) a cost-aware acquisition module. **The aggregation module** can collect multiple responses for the same instance and use them to jointly infer final labels and model reliability. Repeated calls from the same model on the same instance share an instance-level confusion row, so they are marginally dependent rather than independent annotators. **The acquisition module** uses the current label and reliability estimates to decide which model to query for which instance, directing the fixed budget toward the queries expected to be most valuable. These two components reinforce each other: learned reliability guides subsequent budget allocation, while every newly acquired response improves both label inference and future allocation decisions. CALE therefore turns aggregation and allocation into a coupled process, enabling fine-grained use of a fixed budget across heterogeneous LLM annotators.

Concretely, our contributions are threefold:

1. **A hierarchical dependence-aware ensemble for heterogeneous LLM pools.** We model annotator reliability at both regime and instance levels, allowing similar instances to share statistical strength while preserving instance-specific variation. Repeated calls from the same model and instance are treated as dependent evidence rather than independent annotators.
2. **A closed-loop acquisition–aggregation framework.** We couple sequential acquisition with label aggregation rather than treating them as independent stages. At each round, current label and reliability estimates guide the next model–instance query under the remaining budget; the acquired response then updates both estimates for subsequent decisions. This interaction lets the two modules reinforce each other, improving both label inference and budget allocation.
3. **Empirical validation across tasks and deployment conditions.** Across five benchmarks, CALE achieves the highest AUBC on every dataset and obtains its clearest gains at tight budgets. Ablation and robustness studies identify the contributions of the two modules and evaluate stability across modeling and deployment choices.

The rest of the paper is organized as follows. Section 2 formalizes the problem, Section 3 presents CALE, Section 4 reports experiments, Section 5 reviews related work, and Section 6 concludes.

2 Problem Setup

Fixed-pool budgeted annotation. We consider annotating a fixed pool of unlabeled instances for multi-class text classification. For a positive integer d, let $[d] = \{1, \dots, d\}$. Let

$$\mathcal{Q} = \{x_1, \dots, x_N\}, \qquad \mathcal{Y} = \{1, \dots, C\},$$

where each instance x_i has an unknown label $Y_i \in \mathcal{Y}$. The learner has access to a heterogeneous pool of annotator models

$$\mathcal{G} = \{g_1, \dots, g_M\},$$

which may contain any number of large and small models. Model g can be queried at a known normalized cost $c_g > 0$. Gold labels are never revealed during annotation.

Responses and observed history. At each step, the learner chooses one instance–model pair (i, g) and receives one response (i.e., a label). Repeated calls are allowed because LLM outputs may vary across calls, and each pair can be queried at most V times. Let

$$A_{i,g,v} \in \mathcal{Y}, \qquad v \in [V],$$

where $A_{i,g,v}$ denotes the response that model g would return on call slot v for instance i; $a_{i,g,v}$ denotes the observed response on that call. After t steps, let $\mathcal{V}_{i,g}^t \subseteq [V]$ be the slots used for model g

on instance i, where $V_{i,g}^t = |\mathcal{V}_{i,g}^t|$ denotes its cardinality. The corresponding history is

$$\mathcal{H}_{i,g}^t = \{(v, a_{i,g,v}) : v \in \mathcal{V}_{i,g}^t\}.$$

Let $\mathcal{H}_i^t = \{\mathcal{H}_{i,g}^t : g \in \mathcal{G}\}$ for the history of instance i and $\mathcal{H}^t = \{\mathcal{H}_i^t\}_{i=1}^N$ for the full history.

Budget and admissible actions. The total budget is B. After t steps, the remaining budget is

$$B_t = B - \sum_{\tau<t} c_{g_\tau},$$

where g_τ is the model selected at step τ. The admissible action set is

$$\mathcal{A}^t = \left\{(i,g) : i \in [N],\ g \in \mathcal{G},\ V_{i,g}^t < V,\ c_g \le B_t\right\}.$$

Overall, our goal is to obtain high-quality labels for the fixed pool under the available budget.

3 CALE: Cost-Aware Large–small Ensemble

CALE couples two complementary modules, as illustrated in Figure 1. The hierarchical aggregation module converts the response history into current beliefs about labels and model reliability, sharing information across instances with similar error patterns while allowing instance-specific behavior. The cost-aware acquisition module uses these beliefs to select the next instance–model query by combining two values: immediate improvement of the selected label and information that can improve future reliability estimates. Because each response is added back to the history, the two modules reinforce one another throughout the budget trajectory. The posterior and predictive quantities are Bayesian conditional on the current shared parameters; those parameters are refreshed with an approximate empirical-Bayes update, while the global reliability-learning value is a tractable surrogate. See Appendix A for the notations used in the methodology.

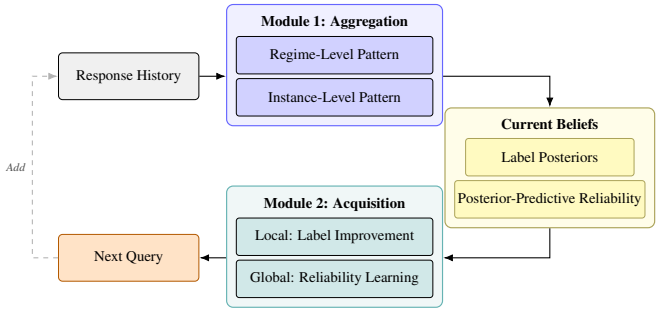

Figure 1: Overview of CALE: the aggregation module converts the response history into current beliefs, which form the interface to the acquisition module. The acquisition module combines local label improvement with global reliability learning to choose the next cost-normalized query; the new response is then added to the history.

The aggregation module specifies the generative model and uses the observed history to form current beliefs. The acquisition module then evaluates affordable candidate responses and selects the next query from those beliefs.

3.1 Aggregation module: hierarchical dependence-aware ensemble

3.1.1 Generative model

Instances may exhibit different model-error patterns. To share information across instances with similar patterns without forcing their behavior to be identical, we associate each instance with a latent regime $Z_i \in \{1, \dots, K\}$. Before annotation begins, the learner assigns a fixed soft prior

$$\rho_{ik} = \mathbb{P}(Z_i = k \mid x_i), \qquad \rho_i = (\rho_{i1}, \dots, \rho_{iK}).$$

The prior ρ_i remains fixed during the episode. A regime is a statistical sharing device, not a hard cluster.

Let $\pi \in \Delta_C$ be the global class prior. Before observing responses, we use the factorization

$$\mathbb{P}(Y_i = y, Z_i = k \mid x_i) = \pi_y \rho_{ik}. \tag{3.1}$$

This is a pre-response prior assumption; it does not imply that Y_i and Z_i remain independent after responses are observed. The current formulation uses x_i to construct the regime prior but does not separately model a feature-conditioned class prior.

For model g and regime k, let $\mathbf{\Psi}_{g,k} \in (\Delta_C^\circ)^C$ be a regime-level confusion matrix. For each instance–model pair, let $\mathbf{\Phi}_{i,g} \in (\Delta_C^\circ)^C$ be an instance-level confusion matrix. Each matrix has one row for each possible true class. We write

$$\boldsymbol{\psi}_{g,k,y} = (\psi_{g,k,y,1}, \ldots, \psi_{g,k,y,C}), \qquad \boldsymbol{\phi}_{i,g,y} = (\phi_{i,g,y,1}, \ldots, \phi_{i,g,y,C}),$$

where y is the true class and c is the emitted response class. The notation $(\Delta_C^\circ)^C$ means that every row is a strictly positive distribution over the C response classes.

The two levels are linked by

$$\boldsymbol{\phi}_{i,g,y} \mid Z_i = k \sim \mathrm{Dir}\big(\kappa \boldsymbol{\psi}_{g,k,y}\big), \qquad g \in \mathcal{G},\ y \in \mathcal{Y}, \tag{3.2}$$

where $\kappa > 0$ controls the strength of shrinkage toward the regime-level row. Large κ keeps instance behavior close to the regime average, whereas small κ allows larger instance-specific deviations. We keep κ fixed throughout an episode.

The generative process is

$$Y_i \sim \mathrm{Cat}(\pi), \qquad Z_i \mid x_i \sim \mathrm{Cat}(\rho_i), \tag{3.3}$$

$$\boldsymbol{\phi}_{i,g,y} \mid Z_i = k \sim \mathrm{Dir}(\kappa \boldsymbol{\psi}_{g,k,y}), \qquad A_{i,g,v} \mid \mathbf{\Phi}_{i,g}, Y_i \sim \mathrm{Cat}(\boldsymbol{\phi}_{i,g,Y_i}). \tag{3.4}$$

Let $\Theta = (\pi, \{\mathbf{\Psi}_{g,k}\}_{g \in \mathcal{G},\, k \in [K]})$ denote the estimated global parameters; κ is fixed and is not part of Θ. The true label selects the active row at the response step; it is not a parent of the whole matrix $\mathbf{\Phi}_{i,g}$. Repeated responses share this row: they are conditionally independent given the row but become dependent after the row is integrated out. Different models are conditionally independent given (Z_i, Θ), while residual correlations from shared training data or prompts are not modeled. As $\kappa \to \infty$, the instance rows collapse to their regime means and the extra-multinomial dependence disappears. Appendix B states these assumptions formally.

The generative process is illustrated in Figure 2.

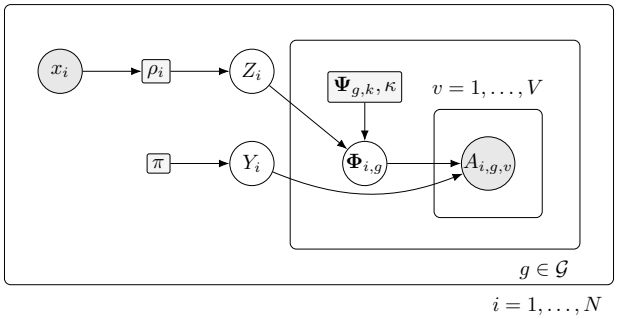

Figure 2: Illustration of the hierarchical aggregation model. The regime index k on $\mathbf{\Psi}_{g,k}$ is selected through Z_i; a separate plate over the K shared regime parameters is omitted for visual clarity.

3.1.2 Model inference

Let $\widehat{\Theta}^{(t)}$ denote the current estimates of the global parameters. The concentration κ remains fixed. Define the response counts

$$n_{i,g,c}^{(t)} = \big|\{v \in \mathcal{V}_{i,g}^t : a_{i,g,v} = c\}\big|, \qquad \mathbf{n}_{i,g}^{(t)} = (n_{i,g,1}^{(t)}, \ldots, n_{i,g,C}^{(t)}).$$

Integrating out the shared instance row gives the count-based, exchangeable Dirichlet–compound–multinomial likelihood

$$L_{i,g}^{(t)}(y,k) = \frac{\mathrm{B}\big(\kappa \widehat{\boldsymbol{\psi}}_{g,k,y}^{(t)} + \mathbf{n}_{i,g}^{(t)}\big)}{\mathrm{B}\big(\kappa \widehat{\boldsymbol{\psi}}_{g,k,y}^{(t)}\big)}. \tag{3.5}$$

Here $\mathrm{B}(\boldsymbol{\alpha}) = \prod_c \Gamma(\alpha_c)/\Gamma(\sum_c \alpha_c)$ is the multivariate beta function. The likelihood depends on response counts rather than their order; the usual multinomial coefficient is omitted because it is constant in (y, k). If model g has not been queried on instance i, set $L_{i,g}^{(t)}(y, k) = 1$.

The joint posterior belief is

$$b_i^{(t)}(y, k) = \mathbb{P}(Y_i = y, Z_i = k \mid \mathcal{H}_i^t, x_i, \widehat{\Theta}^{(t)}) \propto \widehat{\pi}_y^{(t)} \rho_{ik} \prod_{g \in \mathcal{G}} L_{i,g}^{(t)}(y, k), \tag{3.6}$$

normalized over (y, k). Marginalizing the regime gives the posterior label distribution:

$$p_i^{(t)}(y) = \sum_{k=1}^{K} b_i^{(t)}(y, k).$$

The corresponding prediction and posterior decision risk are

$$\hat{y}_i^{(t)} = \arg\max_{y \in \mathcal{Y}} p_i^{(t)}(y), \qquad r_i^{(t)} = 1 - p_i^{(t)}\big(\hat{y}_i^{(t)}\big).$$

These quantities are inferred without access to gold labels.

Conditioned on $(Y_i = y, Z_i = k)$, conjugacy gives

$$\boldsymbol{\phi}_{i,g,y} \mid \mathcal{H}_{i,g}^t, y, k \sim \mathrm{Dir}\big(\kappa \widehat{\boldsymbol{\psi}}_{g,k,y}^{(t)} + \mathbf{n}_{i,g}^{(t)}\big).$$

Therefore, the predictive probability that model g emits class c on one additional call is

$$\bar{\phi}_{i,g,k,y,c}^{(t)} = \frac{\kappa \widehat{\psi}_{g,k,y,c}^{(t)} + n_{i,g,c}^{(t)}}{\kappa + V_{i,g}^t}. \tag{3.7}$$

Before any response from (i, g), this reduces to the regime-level row; after repeats it moves toward the empirical response distribution at a rate controlled by κ.

Every R observed responses, we refresh the class prior and shared regime rows. Let $\nu > 0$ be the total concentration of a symmetric class-prior Dirichlet distribution, and let $\alpha_{y,c} > 0$ be diagonal-biased pseudo-counts. The approximate empirical-Bayes update is

$$\widehat{\pi}_y^{(t)} = \frac{\nu/C + \sum_i p_i^{(t)}(y)}{\nu + N}, \qquad \widehat{\psi}_{g,k,y,c}^{(t)} = \frac{\alpha_{y,c} + \sum_i b_i^{(t)}(y, k) n_{i,g,c}^{(t)}}{\sum_{c'} \left[\alpha_{y,c'} + \sum_i b_i^{(t)}(y, k) n_{i,g,c'}^{(t)}\right]}. \tag{3.8}$$

For clarity, the unnormalized pseudo-count state is

$$\alpha_{\mathrm{EM},g,k,y,c}^{(t)} = \alpha_{y,c} + \sum_i b_i^{(t)}(y, k) n_{i,g,c}^{(t)}, \qquad \widehat{\boldsymbol{\psi}}_{g,k,y}^{(t)} = \frac{\alpha_{\mathrm{EM},g,k,y,\cdot}^{(t)}}{\sum_{c'} \alpha_{\mathrm{EM},g,k,y,c'}^{(t)}}. \tag{3.9}$$

This refresh is a tractable pseudo-count approximation to the finite-κ DCM M-step, not an exact finite-κ EM update. The DCM likelihood and predictive rows remain exact for the current plug-in parameters.

3.2 Acquisition module: cost-aware one-step selection

A query can improve the current instance's label estimate or provide information that helps later queries on other instances. Optimizing over all future query sequences is intractable, so the acquisition module evaluates each affordable pair one step ahead and combines these two values in a cost-normalized score.

3.2.1 Candidate-response prediction

For an affordable pair (i, g) and candidate response $a \in \mathcal{Y}$, the predictive distribution is

$$p_{i,g}^{(t)}(a) = \sum_{y,k} b_i^{(t)}(y, k) \bar{\phi}_{i,g,k,y,a}^{(t)}. \tag{3.10}$$

If the response were a, the hypothetical posterior would be

$$b_i^{(t,+a)}(y, k) \propto b_i^{(t)}(y, k) \bar{\phi}_{i,g,k,y,a}^{(t)}, \tag{3.11}$$

normalized over (y, k).

3.2.2 LOCAL LABEL-IMPROVEMENT VALUE

The hypothetical posterior induces $p_i^{(t,+a)}(y) = \sum_k b_i^{(t,+a)}(y,k)$ and $r_i^{(t,+a)} = 1 - \max_y p_i^{(t,+a)}(y)$. The local gain is

$$\Delta_{\text{local}}^{(t)}(i,g) = r_i^{(t)} - \sum_{a \in \mathcal{Y}} p_{i,g}^{(t)}(a) r_i^{(t,+a)}. \tag{3.12}$$

Under frozen global parameters, executing (i,g) changes only instance i's posterior, so this is the local approximation to pool-risk reduction up to the common factor $1/N$. It measures expected label improvement, not gold-label accuracy. Because the next response becomes more predictable as repeats accumulate, the local gain can decay without an external repeat penalty.

3.2.3 GLOBAL RELIABILITY-LEARNING VALUE

A query can also help estimate a shared regime-level reliability row, even when it is unlikely to change the current label. The acquisition module uses a tractable entropy-based value to represent this future benefit; it is not an exact Bayesian information gain. With current responsibilities held fixed, let α_{score} be a scoring-only copy of the pseudo-count state. It is initialized from α_{EM} and reset to that state after each refresh, so hypothetical scoring updates do not change inference parameters. Let $H_{g,k,y}^{(t)}$ denote the entropy of its current Dirichlet row. For a candidate response a, define

$$\begin{aligned}
\Delta_{\text{param}}^{(t)}(i,g) &= \sum_{y,k} b_i^{(t)}(y,k) \Big[H_{g,k,y}^{(t)} - \sum_{a \in \mathcal{Y}} \bar{\phi}_{i,g,k,y,a}^{(t)} H\Big(\text{Dir}\Big(\alpha_{\text{score},g,k,y,\cdot}^{(t)} + \delta\alpha_{i,g,k,y,\cdot}(a) \Big) \Big) \Big], \\
\delta\alpha_{i,g,k,y,c}(a) &= \frac{\kappa}{\kappa + V_{i,g}^t} \Big(\mathbf{1}[c = a] - \bar{\phi}_{i,g,k,y,c}^{(t)} \Big).
\end{aligned} \tag{3.13}$$

The factor $\kappa/(\kappa + V_{i,g}^t)$ gives the largest reliability-learning signal to a first query and shrinks it under repeats. The perturbation is zero-sum across response classes, so it changes the shape of a row rather than its total concentration. This is an exploration surrogate, not an exact Bayesian information gain. After observing response a_t, update the scoring state only for the selected pair:

$$\alpha_{\text{score},g,k,y,c}^{(t+1)} = \alpha_{\text{score},g,k,y,c}^{(t)} + b_i^{(t)}(y,k) \delta\alpha_{i,g,k,y,c}(a_t). \tag{3.14}$$

3.2.4 COST-NORMALIZED QUERY SELECTION

Finally, CALE selects the admissible pair with the largest cost-normalized score

$$\begin{aligned}
U^{(t)}(i,g) &= \frac{\Delta_{\text{local}}^{(t)}(i,g) + \beta [\Delta_{\text{param}}^{(t)}(i,g)]_+ / \bar{H}_0}{c_g}, \\
\bar{H}_0 &= \frac{1}{MKC} \sum_{g \in \mathcal{G}} \sum_{k \in [K]} \sum_{y \in \mathcal{Y}} \Big| H\Big(\text{Dir}\Big(\alpha_{\text{score},g,k,y,\cdot}^{(0)} \Big) \Big) \Big|.
\end{aligned} \tag{3.15}$$

Here $\beta \geq 0$ controls the relative weight of reliability learning, $[u]_+ = \max(u, 0)$, and c_g is the normalized query cost. The positive-part operation prevents a negative entropy surrogate from reducing the score. CALE queries the admissible pair with the largest $U^{(t)}$, refreshes the shared estimates every R responses, and after the terminal step T outputs $\hat{y}_i^{(T)} = \arg\max_{y \in \mathcal{Y}} p_i^{(T)}(y)$. The complete pseudocode, positivity handling, and implementation details are given in Appendix C–D.

4 EXPERIMENTS

We evaluate CALE in three ways. First, we compare its anytime accuracy–cost trade-off with budgeted acquisition and model-routing baselines (Section 4.2). Second, we isolate the contribution of each component (Section 4.3). Third, we evaluate sensitivity to modeling and deployment choices (Section 4.4).

4.1 SETUP

Datasets. We annotate five datasets spanning distinct task types: AG News (Zhang et al., 2015) for news topic classification, MedNLI (Romanov & Shivade, 2018) for clinical inference, Recognizing

Textual Entailment (RTE) (Wang et al., 2018) for textual entailment, DBPedia-14 (Zhang et al., 2015) for 14-way ontology classification, and Massive Multitask Language Understanding (MMLU) (Hendrycks et al., 2021) for multi-subject knowledge questions. Appendix E gives sizes, label spaces, and the sampling protocol.

Annotator pool and costs. The annotator pool holds three small models (`deepseek-r1:14b`, `phi4:14b`, `qwen2.5:14b`) at $c_S{=}1$ and one large model (`gemma4:31b`) at $c_L{=}10$, each queryable $V{=}3$ times per instance with temperature sampling (Wang et al., 2023). We report a normalized cost ratio rather than a vendor price list, because absolute prices change frequently and self-hosted costs are hardware-specific. Every run uses $B{=}4N$ and ten seeds.

Metrics. We report two metrics. AUBC, the area under the accuracy–cost curve over $[0, B]$ normalized by B, which scores a policy across the whole budget rather than at a single operating point. Final accuracy at $B{=}4N$ is also reported, although by that budget most policies have converged and the endpoint compresses the differences a cost-aware method should produce.

Baselines. We compare against two groups of baselines. The simple policies *Random*, *Cost-weighted Random*, and *Cheap-First* set the scale with Dawid–Skene (DS) (Dawid & Skene, 1979), while the primary system comparison is against *DLTA** (Zheng & Chen, 2018), *LMCrowd* (Zhang et al., 2025), and *QUORUM* (Purificato et al., 2026). The aggregation ablation also includes GLAD (Generative model of Labels, Abilities, and Difficulties) (Whitehill et al., 2009). The star in DLTA* denotes our heterogeneous-cost adaptation, which weights entropy reduction per unit of money (Appendix F).

4.2 Main results

Table 1: Main results: AUBC (%), reported as mean±sd over ten seeds. We use two-sided Wilcoxon signed-rank tests to compare CALE with each baseline. Tests are performed over the ten seeds for each dataset and over all 50 dataset–seed cells for the Avg. column. Stars indicate $^{***}p < 0.01$, $^{**}p < 0.05$, and $^{*}p < 0.1$. Among the system-level methods, CALE achieves the highest AUBC on every dataset and the highest overall mean.

Method	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
Random	50.16±0.83***	53.78±0.72***	67.04±0.39***	45.80±0.76***	44.44±0.74***	52.24***
Cost-weighted Random	65.50±0.55***	65.08±0.78***	77.54±0.42***	58.76±0.48***	67.99±0.64***	66.98***
Cheap-First + DS	67.82±0.36***	66.39±0.43***	78.75±0.39***	60.59±0.38***	70.58±0.54***	68.83***
QUORUM	58.00±0.01***	59.39±0.01***	72.69±0.01***	54.72±0.01***	55.80±0.01***	60.12***
LMCrowd	71.92±0.26***	73.48±0.19***	84.29±0.12***	66.96±0.25***	83.83±0.10***	76.10***
DLTA*	78.24±0.37***	72.93±0.17***	85.36±1.04*	66.36±1.48***	84.28±0.47***	77.43***
CALE (ours)	**78.83**±0.24	**74.48**±0.26	**86.02**±0.65	**71.30**±0.22	**84.90**±0.12	**79.11**

Table 1 shows that CALE achieves the highest AUBC on all five datasets and the highest mean (79.11), ahead of DLTA* (77.43) and LMCrowd (76.10). The margin over the strongest competing system is largest on MMLU (+4.34 points) and smallest on AG News (+0.59). QUORUM scores 60.12, but its AUBC interpolates independent fixed-budget runs, so that comparison is protocol-qualified.

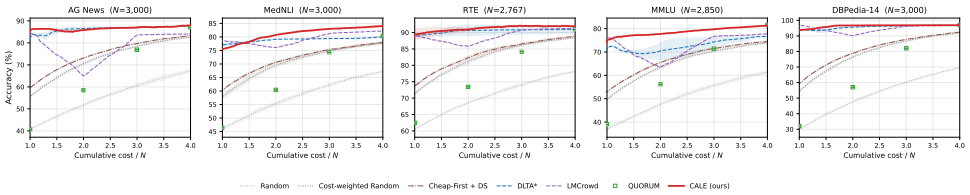

Figure 3: Accuracy against cumulative cost, one panel per dataset. Each curve is the mean over the ten seeds, and the shaded region around it spans one standard deviation on either side. QUORUM is drawn as markers because each of its points is an independent fixed-budget run, so the segments between them are not a path it traverses.

Figure 3 shows where the gain arises. CALE initially trails LMCrowd, overtakes it at about $1.12N$, and reaches 85.5% mean accuracy at $1.5N$, whereas DLTA* and LMCrowd require about $2.91N$

and $2.98N$. By $4N$, CALE reaches 88.4%, compared with about 86.4% for both baselines. Thus the main gain is achieving a given accuracy sooner, not a large change in the final destination.

The allocation pattern supports this interpretation. The local gain prioritizes queries that may settle the current label, while the global surrogate values queries that may improve reliability estimates for later decisions. Across the tested positive exploration weights, mean AUBC changes by only 0.15 points(Appendix G.2). Under the $10{:}1$ cost ratio, CALE spends 31.8% of its budget on the large model and uses repeats for only 11.7% of its calls, compared with 25.0% and 11.7% for LMCrowd and 10.5% and 44.6% for DLTA*. The dependence-aware predictive model discounts another response from the same model–instance pair; the resulting allocation is consistent with spending more on informative model choices and less on redundant votes.

4.3 Ablation studies

We isolate the hierarchy, acquisition rule, and aggregation estimator. Main text reports Final Accuracy at $B{=}4N$; AUBC companions are given in Appendix G.1.

Table 2: Hierarchy ablation, Final Accuracy (%, mean±sd over seeds).

Variant	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
No regime	86.47±0.87	79.10±0.11	89.82±2.11	76.20±0.17	95.76±0.76	85.47
Regime only	86.64±0.06	78.91±0.42	88.79±3.22	72.38±3.50	95.35±1.46	84.41
Instance only	86.26±0.99	83.90±0.39	86.89±0.38	**82.21**±0.44	**96.97**±0.42	87.25
CALE	**87.97**±0.43	**84.03**±0.62	**91.96**±0.16	81.32±0.51	96.87±0.14	**88.43**

Regime structure. Table 2 removes each level of the hierarchy. Keeping neither level costs 2.96 points and keeping only the regime level costs 4.02, so neither level alone reproduces the full model. The instance level carries most of the signal, and on MMLU — where repeated same-model calls are most correlated — removing it costs 5.1 and 8.9 points. The finite-K regime mixture then adds 1.18 points on top of the instance level ($p{=}7.6\mathrm{e}{-3}$), a smaller average contribution that comes from pooling reliability evidence across instances rather than estimating every instance independently.

Table 3: Acquisition ablation: the acquisition rule varies, CALE's aggregator is held fixed. Final Accuracy (%, mean±sd over seeds).

Acquisition rule	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
BALD / cost	87.41±0.28	79.99±0.51	91.23±0.32	77.23±0.39	96.61±0.39	86.49
Entropy / cost	86.87±0.33	46.93±6.63	87.10±12.86	80.38±1.38	89.55±0.13	78.17
AccEIG / cost	86.64±0.60	79.98±0.73	91.06±0.37	77.68±0.32	96.63±0.22	86.40
Proactive	87.31±0.31	80.99±0.55	91.41±0.14	77.19±0.34	96.48±0.31	86.68
CALE	**87.97**±0.43	**84.03**±0.62	**91.96**±0.16	**81.32**±0.51	**96.87**±0.14	**88.43**

Acquisition. Table 3 attributes about 1.8 points to the dual-gain rule: the best alternative reaches 86.68 versus 88.43 for CALE, with the largest gaps on MedNLI and MMLU. Entropy/cost concentrates 81% of its budget on one model family and leaves 89% of RTE instances single-sourced, whereas CALE leaves at most 6.5% single-sourced.(Table 12)

Table 4: Aggregation ablation: the aggregator varies over one shared CALE acquisition episode. Final Accuracy (%, mean±sd over seeds).

Aggregator	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
Majority vote	87.19±0.20	83.42±0.45	92.14±0.30	80.05±0.49	96.80±0.15	87.92
Dawid–Skene ($K{=}1$)	**88.02**±0.43	83.28±0.57	91.86±0.15	79.08±0.59	96.85±0.15	87.82
GLAD	87.71±0.33	82.13±0.43	**92.32**±0.18	80.14±0.35	96.82±0.15	87.82
CALE	87.97±0.43	**84.03**±0.62	91.96±0.16	**81.32**±0.51	**96.87**±0.14	**88.43**

Aggregation. Table 4 runs one acquisition episode and varies only the aggregator. With the acquisition history fixed, aggregation variants range from 87.82 to 88.43. CALE improves over majority vote and Dawid–Skene by 0.51 and 0.61 points, respectively, with the largest gains on MMLU. It also outperforms GLAD, which models difficulty as one scalar per instance, whereas

CALE's instance-level rows $\phi_{i,g}$ let an instance be hard for one model but not another, and the regimes share these difficulty patterns across instances. Acquisition contributes more at this budget, but dependence-aware aggregation remains necessary to exploit the acquired responses.

4.4 Robustness

Feature-based pre-clustering can be noisy, so we fix $K{=}3$. Mean AUBC varies by only 0.11 points across the tested K values and 0.15 points across positive β values. Finite κ values are stable, whereas $\kappa \to \infty$ removes repeated-call dependence and lowers mean AUBC by 2.43 points.

The advantage persists under deployment changes: changing R from 10 to 60 changes mean AUBC by only 0.07 points, and CALE remains best across the tested cost ratios and model pools. Detailed sweeps are reported in Appendix G.2. Scoring all admissible actions costs $O(|\mathcal{A}^t|KC^2)$ per acquisition step, with a median runtime of about 8 minutes per main or ablation cell. Appendix C.2 gives the detailed complexity and runtime analysis.

5 Related Work

Cost-Aware Acquisition. Budgeted active learning and crowdsourcing choose the next instance–annotator query using uncertainty, information gain, and annotator cost or reliability (Yan et al., 2011; Ho et al., 2013; Frazier et al., 2008; Chen et al., 2015; Golovin & Krause, 2011). Recent LLM systems extend this quality–cost trade-off to heterogeneous model pools through contextual-bandit selection, selective relabeling, or budget-constrained model subsets (Elumar et al., 2026; Zhang et al., 2025; Huang et al., 2025). Routing and cascading methods instead choose a model or escalation path for each request, often using predicted quality, confidence, preference data, or a trained router (Shnitzer et al., 2023; Ding et al., 2024; Zhang et al., 2024; Dekoninck et al., 2024; Ong et al., 2025; Maurya et al., 2025). Unlike these per-request decisions, CALE allocates one global budget over a fixed unlabeled pool and learns model–instance reliability from acquired responses, so acquisition can exploit complementary error patterns without gold labels or auxiliary training.

Crowdsourcing Aggregation. Learning-from-crowds methods infer latent labels and annotator reliability through confusion matrices and EM- or Bayesian-style inference, with Dawid–Skene and its extensions providing the standard foundation (Dawid & Skene, 1979; Whitehill et al., 2009; Raykar et al., 2010; Kim & Ghahramani, 2012; Moreno et al., 2015). Models of redundant annotations and worker correlation further show that votes need not be independent when annotators share error patterns (Li et al., 2019a;b). Recent LLM annotation systems apply voting, truth inference, selective verification, or human–LLM allocation to combine machine-generated labels (Farr et al., 2024; He et al., 2024; Zhang et al., 2025; Li et al., 2023). CALE extends this line by treating repeated calls from the same model on the same instance as dependent through a shared instance-level confusion row, while regime-level priors transfer reliability across similar instances; the resulting posterior also determines the next acquisition, coupling aggregation and budget allocation.

6 Concluding Remarks

CALE couples dependence-aware aggregation with cost-normalized acquisition for budgeted annotation without gold labels during execution. Its hierarchical model shares reliability information across instances while accounting for dependence among repeated calls. Across five benchmarks, CALE achieves the highest AUBC on every dataset, with the clearest gains at tight budgets; ablations and robustness studies explain the source and stability of this improvement.

This work suggests several future directions. The framework could be extended to more complex, structured annotation tasks, such as span extraction, multi-step reasoning, and summarization, where outputs are structured or generative rather than single class labels. Other directions include larger and more diverse model pools, multilingual or multimodal data, and richer dependence models that capture shared errors across models. Deployments in high-stakes settings should also examine bias amplification and systematic LLM errors alongside annotation quality and cost.

AI USE STATEMENT

We used generative AI tools only to assist with language editing and code-documentation tasks. All scientific claims, experimental results, mathematical derivations, and final text were reviewed and verified by the authors, who take responsibility for the submitted work.

ETHICS STATEMENT

This work studies automated data annotation and does not involve human subjects or the collection of private personal data. Potential biases in model-generated annotations remain an important limitation of this setting; the paper therefore treats model reliability as an empirical quantity rather than assuming that any single model is universally correct.

REPRODUCIBILITY STATEMENT

The problem formulation, modeling assumptions, acquisition rule, and algorithm are described in Sections 2, 3, and Appendix D. Experimental protocols, datasets, model pools, budgets, and additional sensitivity analyses are reported in Section 4 and the appendix.

REFERENCES

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Xi Chen, Qihang Lin, and Dengyong Zhou. Statistical decision making for optimal budget allocation in crowd labeling. *The Journal of Machine Learning Research*, 16(1):1–46, 2015.

Alexander Philip Dawid and Allan 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.

Jasper Dekoninck, Maximilian Baader, and Martin Vechev. A unified approach to routing and cascading for llms. *arXiv preprint arXiv:2410.10347*, 2024.

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 (Volume 1: Long Papers)*, pp. 11173–11195, 2023.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Rühle, Laks Lakshmanan, and Ahmed H Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. In *International Conference on Learning Representations*, volume 2024, pp. 41348–41366, 2024.

Eray Can Elumar, Cem Tekin, and Osman Yagan. Cost-aware llm-based online dataset annotation. *Advances in Neural Information Processing Systems*, 38:3516–3559, 2026.

David Farr, Nico Manzonelli, Iain Cruickshank, Kate Starbird, and Jevin West. Llm chain ensembles for scalable and accurate data annotation. In *2024 IEEE International Conference on Big Data (BigData)*, pp. 2110–2118. IEEE, 2024.

Peter I Frazier, Warren B Powell, and Savas Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47(5):2410–2439, 2008.

Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

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 the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pp. 165–190, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Chien-Ju Ho, Shahin Jabbari, and Jennifer Wortman Vaughan. Adaptive task assignment for crowdsourced classification. In *International conference on machine learning*, pp. 534–542. PMLR, 2013.

Keke Huang, Yimin Shi, Dujian Ding, Yifei Li, Yang Fei, Laks Lakshmanan, and Xiaokui Xiao. Thriftllm: On cost-effective selection of large language models for classification queries. *arXiv preprint arXiv:2501.04901*, 2025.

Hyun-Chul Kim and Zoubin Ghahramani. Bayesian classifier combination. In *Artificial Intelligence and Statistics*, pp. 619–627. PMLR, 2012.

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, 2023.

Yuan Li, Benjamin IP Rubinstein, and Trevor Cohn. Truth inference at scale: A bayesian model for adjudicating highly redundant crowd annotations. In *The World Wide Web Conference*, pp. 1028–1038, 2019a.

Yuan Li, Benjamin Rubinstein, and Trevor Cohn. Exploiting worker correlation for label aggregation in crowdsourcing. In *International conference on machine learning*, pp. 3886–3895. PMLR, 2019b.

Kaushal Kumar Maurya, KV Aditya Srivatsa, and Ekaterina Kochmar. Selectllm: Query-aware efficient selection algorithm for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20847–20863, 2025.

P Moreno, Antonio Artés-Rodríguez, Y Teh, and Fernando Perez-Cruz. Bayesian nonparametric crowdsourcing. *Journal of Machine Learning Research*, 2015.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Antonio Purificato, Maria Sofia Bucarelli, Andrea Bacciu, Amin Mantrach, and Fabrizio Silvestri. Quorum: Quality-optimized routing using multiple annotators. *arXiv preprint arXiv:2608.27974*, 2026.

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(4), 2010.

Alexey Romanov and Chaitanya Shivade. Lessons from natural language inference in the clinical domain. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 1586–1596, 2018.

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*, 2023.

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, 2024.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. Want to reduce labeling cost? gpt-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4195–4205, 2021.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jacob Whitehill, Ting-fan Wu, Jacob Bergsma, Javier Movellan, and Paul Ruvolo. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. *Advances in neural information processing systems*, 22, 2009.

Yan Yan, Glenn M Fung, Rómer Rosales, and Jennifer G Dy. Active learning from crowds. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 1161–1168, 2011.

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, 2025.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Xuechen Zhang, Zijian Huang, Ege Onur Taga, Carlee Joe-Wong, Samet Oymak, and Jiasi Chen. Efficient contextual llm cascades through budget-constrained policy learning. *Advances in Neural Information Processing Systems*, 37:91691–91722, 2024.

Libin Zheng and Lei Chen. Dlta: A framework for dynamic crowdsourcing classification tasks. *IEEE Transactions on Knowledge and Data Engineering*, 31(5):867–879, 2018.

A NOTATION

This section collects the notation used in our study.

Symbol	Meaning
Sets, indices, and task interface	
$[d]$	The index set $\{1, \dots, d\}$.
$\mathcal{Q} = \{x_i\}_{i=1}^N$	Fixed pool of N unlabeled instances.
$\mathcal{Y} = [C]$	Label space with C possible classes; $Y_i \in \mathcal{Y}$ is the unknown label of instance i.
$\mathcal{G} = \{g_1, \dots, g_M\}$	Pool of M annotator models; $c_g > 0$ is the normalized cost of querying model g.
V	Maximum number of response slots for one instance–model pair.
$A_{i,g,v}$, $a_{i,g,v}$	Potential label returned at slot v and the observed label at that slot, respectively.
History, budget, and counts	
$\mathcal{V}_{i,g}^t$, $V_{i,g}^t$	Used response slots and their count for pair (i, g) at step t.
$\mathcal{H}_{i,g}^t$, $\mathcal{H}_i^t$, $\mathcal{H}^t$	History for one pair, one instance, and the full pool, respectively.
B, B_t	Total budget and remaining budget after t responses.
$\mathcal{A}^t$	Admissible instance–model pairs at step t.
$n_{i,g,c}^{(t)}$, $\mathbf{n}_{i,g}^{(t)}$	Count of response class c and the vector of response counts for pair (i, g).
Prior and hierarchical response model	
Z_i, K, ρ_{ik}	Latent regime, number of regimes, and fixed prior probability $\mathbb{P}(Z_i = k \mid x_i)$.
π	Global prior distribution over true classes.
$\boldsymbol{\Psi}_{g,k}$, $\boldsymbol{\psi}_{g,k,y}$	Regime-level confusion matrix for model g and regime k, and its row for true class y.
$\boldsymbol{\Phi}_{i,g}$, $\boldsymbol{\phi}_{i,g,y}$	Instance-level confusion matrix and its row for instance–model pair (i, g) and true class y.
κ	Fixed Dirichlet concentration linking an instance-level row to its regime-level row.
Θ, $\widehat{\Theta}^{(t)}$	Global model parameters and their current plug-in estimates; κ is fixed and is not included in Θ.
$L_{i,g}^{(t)}(y, k)$	DCM likelihood contribution of pair (i, g) under label–regime state (y, k).
$b_i^{(t)}(y, k)$	Joint posterior belief $\mathbb{P}(Y_i = y, Z_i = k \mid \mathcal{H}_i^t, x_i, \widehat{\Theta}^{(t)})$.
$p_i^{(t)}(y)$	Posterior probability of label y, obtained by summing $b_i^{(t)}(y, k)$ over regimes.
$r_i^{(t)}$, $\hat{y}_i^{(t)}$	Posterior decision risk and corresponding maximum-posterior label.
$\bar{\phi}_{i,g,k,y,c}^{(t)}$	Posterior-predictive probability that model g emits c for instance i.
α_{EM}, α_{score}	Inference pseudo-count state and scoring-only copy used for the reliability-learning surrogate.
$\Delta_{\mathrm{local}}^{(t)}$, $\Delta_{\mathrm{param}}^{(t)}$	Expected local label-improvement gain and global reliability-learning surrogate.
$\delta\alpha_{i,g,k,y,c}(a)$	Hypothetical residual pseudo-count change from candidate response a.
$U^{(t)}(i, g)$, β, $\bar{H}_0$	Cost-normalized query score, reliability-learning weight, and entropy normalizer.

Table 5: Notation used in our study.

B MODELING ASSUMPTIONS

Assumption 1 (Label-regime prior independence)**.** *The true label and the latent regime are a priori independent given instance features:*

$$\mathbb{P}(Y_i = y, Z_i = k \mid x_i) = \pi_y \rho_{ik}. \tag{B.1}$$

This is a prior factorization before any model response is observed. Instance features enter through the regime-routing prior ρ_i, while class prevalence is represented by the global prior π. Observed responses subsequently induce posterior dependence between Y_i and Z_i.

Assumption 2 (Within-model correlation via instance-level confusion)**.** *Conditioned on the instance-level confusion matrix $\boldsymbol{\Phi}_{i,g}$, repeated responses from the same model on the same instance are*

independent categorical draws from the row selected by Y_i:

$$\mathbb{P}(\{A_{i,g,v}\}_{v\in\mathcal{V}^t_{i,g}} \mid \mathbf{\Phi}_{i,g}, Y_i, Z_i, \Theta) = \prod_{v\in\mathcal{V}^t_{i,g}} \mathbb{P}(A_{i,g,v} \mid \boldsymbol{\phi}_{i,g,Y_i}). \tag{B.2}$$

The instance-level confusion matrix is latent and shared by all repeated responses from model g on instance i. Dependence therefore appears after marginalizing $\mathbf{\Phi}_{i,g}$.

Assumption 3 (Cross-model conditional independence)**.** *Conditional on the latent regime and global model parameters, the instance-level confusion matrices of different models are independent:*

$$\mathbb{P}(\{\mathbf{\Phi}_{i,g}\}_{g\in\mathcal{G}} \mid Z_i, \Theta) = \prod_{g\in\mathcal{G}} \mathbb{P}(\mathbf{\Phi}_{i,g} \mid Z_i, \Theta). \tag{B.3}$$

This factorization does not rule out shared instance difficulty, which is carried by Z_i. It rules out residual dependence among different models after conditioning on Z_i. If models share training data, prompting artifacts, or systematic reasoning biases, posterior confidence from cross-model agreement may still be overstated.

C Mathematical and algorithmic details

C.1 Inference and scoring details

Dirichlet entropy. The differential entropy of a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\lambda})$ with $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_C)$ and $\lambda_0 = \sum_u \lambda_u$ is

$$H[\mathrm{Dir}(\boldsymbol{\lambda})] = \ln \mathrm{B}(\boldsymbol{\lambda}) + (\lambda_0 - C)\psi_0(\lambda_0) - \sum_{u=1}^{C}(\lambda_u - 1)\psi_0(\lambda_u), \tag{C.1}$$

where $\mathrm{B}(\cdot)$ is the multivariate Beta function and $\psi_0(\cdot)$ is the digamma function.

DCM versus Dawid–Skene. Under conditional-independence Dawid–Skene, repeated responses from model g contribute $\prod_c \widehat{\Psi}_{g,k,y,c}^{\,n_{i,g,c}}$. Under the hierarchical model, the same count vector contributes the DCM factor in Eq. (3.5). The posterior-predictive probability of a new response c is $(\kappa\widehat{\Psi}_{g,k,y,c} + n^{(t)}_{i,g,c})/(\kappa + V^t_{i,g})$, so existing same-model responses directly affect the marginal value of an additional repeat. The multinomial coefficient in the DCM mass depends only on observed counts and cancels in posterior ratios over (y, k).

The accuracy consequence of this difference is measured on the current setup by the $\kappa \to \infty$ arm of Appendix G.2, which removes the dependence while keeping everything else fixed.

Empirical-Bayes pseudo-count approximation. The exact DCM M-step for $\mathbf{\Psi}_{g,k,y}$ is nonconjugate and involves digamma fixed-point updates. We instead use the responsibility-weighted pseudo-count update in Eq. (3.8). This update matches the usual Dawid–Skene M-step when no pair is queried more than once and matches the $\kappa \to \infty$ limit. For finite κ with repeats, DCM dependence is still used in the E-step and acquisition score; the pseudo-count update is an empirical-Bayes summary for the regime-level row.

Each refresh runs five EM iterations in the reported experiments. At the initial refresh, $\mathbf{\Psi}$ is initialized by row-normalizing the diagonal-biased prior $\alpha_{y,c}$ and π is uniform; subsequent refreshes warm-start $\mathbf{\Psi}$ from the previous refresh. After the last E-step, α_{EM} is recomputed from the final responsibilities and counts (Eq. (3.9)), and this state initializes α_{score}. The posterior and predictive rows are then recomputed with the final $\widehat{\mathbf{\Psi}}$ and $\widehat{\pi}$.

Residual scoring update. The scoring copy α_{score} is used only for Δ_{param} and never changes the inference estimate $\widehat{\Psi}$. Because the residual update in Eq. (3.13) has negative entries, the implementation replaces entries below 10^{-9} by 10^{-9} before evaluating Dirichlet entropies. Negative values of the entropy-reduction surrogate are clipped in Eq. (3.15) so the global term acts as an exploration bonus.

C.2 Computational cost and runtime

The implementation does not materialize a full $N \times M \times C \times C$ instance-confusion tensor. It stores response counts in $O(NMC)$ memory, slot-availability metadata in $O(NMV)$, label–regime beliefs in $O(NKC)$, and the shared reliability states in $O(MKC^2)$ (at most 2,352 floats per state for $M{=}4$, $K{=}3$, and $C \leq 14$). Between EM refreshes, scoring all admissible actions costs $O(|\mathcal{A}^t|KC^2)$ per acquisition step and updating the selected pair costs $O(KC^2)$. For the global-gain computation, candidates with the same response-count vector are grouped so that the entropy term is evaluated once per distinct count state and reused, without changing the resulting scores. Every R responses, EM scans the observed pool and costs $O(NMKC^2)$, or $O(NMKC^2/R)$ amortized per acquired response. The sweep in Appendix G.2 changes mean AUBC by only 0.07 points between $R{=}10$ and $R{=}60$. Cached model responses are generated once; median wall-clock times are about 8 minutes per main or ablation cell and 3 minutes per robustness-sweep cell. The SLURM allocation and hardware are summarized in Table 7.

D Algorithm

Algorithm 1 CALE: Cost-aware acquisition with local and global gains

Require: Budget B, exploration weight β, max repeats V, refresh period R, optional bootstrap history $\mathcal{H}^0$

Initialize $\widehat{\Theta}^{(0)}$ and $\alpha_{\mathrm{EM}}^{(0)}$ by EM on $\mathcal{H}^0$; set $\alpha_{\mathrm{score}}^{(0)} \leftarrow \alpha_{\mathrm{EM}}^{(0)}$
Deduct bootstrap cost from B and compute $b_i^{(0)}$, $\bar{\phi}_{i,g,k,y,c}^{(0)}$
for $t = 0, 1, 2, \ldots$ **do**
    Form $\mathcal{A}^t = \{(i,g) : |\mathcal{V}_{i,g}^t| < V,\ c_g \leq B_t\}$
    if $\mathcal{A}^t = \emptyset$ **then break**
    end if
    Compute $\Delta_{\mathrm{local}}^{(t)}(i,g)$, $\Delta_{\mathrm{param}}^{(t)}(i,g)$, and $U^{(t)}(i,g)$ for all $(i,g) \in \mathcal{A}^t$
    Query $(i_t^\star, g_t^\star) = \arg\max_{(i,g)\in\mathcal{A}^t} U^{(t)}(i,g)$ and append the observed response to $\mathcal{H}^{t+1}$
    Update remaining budget and apply the residual update to α_{score}
    if R new responses have been acquired since the last refresh **then**
        Run EM, update $\widehat{\Theta}^{(t+1)}$ and $\alpha_{\mathrm{EM}}^{(t+1)}$, and reset $\alpha_{\mathrm{score}}^{(t+1)} \leftarrow \alpha_{\mathrm{EM}}^{(t+1)}$
    end if
    Recompute affected posteriors and posterior-predictive rows
end for
Run a final EM refresh and output $\hat{y}_i = \arg\max_y p_i(y)$

E Datasets

Table 6: Datasets. N is the number of pooled instances annotated and C the label-space size.

Dataset	Source	N	C	Task
AG News	Zhang et al. (2015)	3,000	4	News topic classification
MedNLI	Romanov & Shivade (2018)	3,000	3	Clinical natural language inference
RTE	Wang et al. (2018)	2,767	2	Textual entailment
MMLU	Hendrycks et al. (2021)	2,850	4	Multi-subject knowledge and reasoning
DBPedia-14	Zhang et al. (2015)	3,000	14	Ontology / entity classification

Pool construction. Each dataset contributes one fixed pool of instances (Table 6) that every method and seed annotates. AG News, MedNLI, and DBPedia-14 are subsampled from their training splits with a fixed seed shared across all runs, so the ten acquisition seeds differ only in acquisition randomness. DBPedia-14 is shuffled before truncation because its released split is ordered by class: a prefix without shuffling yields two of the fourteen classes. RTE is used whole, concatenating the GLUE training and validation splits ($2{,}490 + 277$).

MMLU. We sample 50 items per subject from the test split, uniformly without replacement, with a per-subject seed so that adding or removing a subject does not reshuffle the others; all 57 subjects yield a full draw, giving 2,850 items. Each item is rendered as the question followed by its four options on separate lines, and the label space is the four option positions $\{A, B, C, D\}$, so C=4. The subject string is given to the annotators and the embedding encoder as a topic hint, never as a label, and items are pooled with no subject stratification.

Prompts. All datasets use a common instruction asking for a JSON object with a label and a reason. The natural-language-inference datasets (MedNLI, RTE) present a premise and a hypothesis; the text-classification datasets present an optional title and the text; MMLU presents the subject, question, and options. In every case the prompt states the allowed labels explicitly, and a response outside that set is treated as missing rather than mapped to a class.

F Implementation and baseline adaptation

Bootstrap accounting. The acquisition episode may optionally start from a bootstrap history $\mathcal{H}^0$ used to initialize posteriors and EM estimates. Bootstrap queries are ordinary model responses included in $\mathcal{H}^0$; gold labels are never revealed. If $\mathcal{I}_{\text{boot}} = \{(i, g, v) : v \in \mathcal{V}^0_{i,g}\}$ denotes the bootstrap query triples, the initial remaining budget is

$$B_0 = B - \sum_{(i,g,v)\in\mathcal{I}_{\text{boot}}} c_g.$$

All experiments reported in this paper use no warm start: every method begins from an empty history and pays for every response it sees.

LLM inference and cache. The three small models (`deepseek-r1:14b`, `phi4:14b`, `qwen2.5:14b`) and the large model (`gemma4:31b`) are served locally and queried at decoding temperature 0.7; each instance–model pair has V=3 response slots. Responses are cached so that any two policies selecting the same (i, g, v) slot on the same dataset–seed pair receive the same response, which is what makes the comparisons paired rather than merely matched. A small number of slots (14 of 175,404, under 0.01%) have no valid response and are treated as unavailable; they are never filled with a substitute. All acquisition policies resolve slots through one shared per-cell realization, so a seed randomizes which slot of a family is consumed but not which instances exist or in what order they are considered.

Embeddings and regime priors. Embeddings use `nomic-embed-text` reduced to 64 dimensions by principal component analysis (PCA). Soft regime priors are formed from those embeddings by a softmax over cluster distances at unit temperature, once per dataset before annotation begins, and are held fixed for every method and seed.

Hyperparameters. Unless a table states otherwise, CALE uses the Dirichlet concentration κ=0.1, K=3 regimes, exploration weight β=2, EM refresh period R=30, class-prior concentration ν=1, and confusion-matrix prior α_{diag}=3.0, α_{off}=1.1, where $\alpha_{y,c} = \alpha_{\text{diag}}$ for $y = c$ and $\alpha_{y,c} = \alpha_{\text{off}}$ otherwise. Appendix G.2 varies each of the first four in turn. All experiments use the same ten acquisition seeds: 1, 7, 13, 42, 77, 123, 200, 314, 456, and 999.

Evaluation grids. Accuracy is evaluated on the full pool after every acquisition step for the majority-vote, Dawid–Skene, and CALE aggregators. GLAD is roughly 90% of a cell's wall clock, so its curve is thinned to every twelfth evaluation point, retaining the final point and every $0.5N$ budget anchor. Its final accuracy is therefore exact while its AUBC is integrated on a coarser grid; re-integrating the majority-vote column on GLAD's grid shifts its AUBC by between -0.026 and $+0.009$ points, an order of magnitude below the across-seed standard deviation.

Computing resources. Table 7 summarizes the computing resources. The expensive model calls are performed once to build the annotation cache; all acquisition policies and the robustness analysis replay the same cached (i, g, v) responses, so reproducing the policy comparisons from the released artifacts requires only CPU computation.

Table 7: Compute resources used for the reported experiments. A cell is one dataset–seed–method run at one experimental setting. Only the first stage needs a GPU; everything downstream replays the cache on CPU.

Stage	Hardware	Allocation and scale
LLM annotation cache	Local GPU server	One pass per dataset, model and response slot; 175,404 slots in total, never repeated across policies.
Embeddings and regime priors	Local GPU server, CPU post-processing	One pass per dataset for `nomic-embed-text`, then PCA and soft assignment; reused by every method and seed.
Acquisition experiments	SLURM CPU array	1 core and 4–6 GB requested per cell; median wall clock is about 8 minutes for main/ablation cells and 3 minutes for sweeps.
Aggregation and plotting	Local CPU	Post-processing of per-cell summaries into the reported tables and figures; seconds, no model inference.

Baseline adaptation. DLTA (Zheng & Chen, 2018) assumes an anonymous, uniform-cost crowd: it decides *how many* labels a task receives, never *which* worker provides them. Three changes bridge it to a heterogeneous-cost pool. The allocation weight becomes entropy reduction per unit of money, which keeps the objective linear and the original allocation step intact; the per-round budget becomes a spend cap rather than a label count; and a selected allocation is mapped to a family-specific response slot. The round structure, entropy-reduction weight, and aggregator are unchanged; the star in DLTA* marks this extension.

LMCrowd (Zhang et al., 2025) and QUORUM (Purificato et al., 2026) are used as published, each with its native aggregator. Both expose exactly one paid arm, so we evaluate them only on the reference pool rather than on pools with zero or two large models. QUORUM is additionally a fixed-budget method: its calibration phase depends on the supplied budget, so a run at B is not a prefix of a run at $B' > B$. We therefore run it once per budget multiplier and treat the resulting points as independent measurements; its missing-slot fallback is recorded in each summary.

G Robustness analysis and companion metrics

This section first reports the companion metric omitted from the main text and then varies one modeling or deployment choice at a time. The reference configuration uses three small models and one large model, c_L/c_S=10, κ=0.1, K=3, β=2, R=30, B=4N, V=3, and the same ten seeds. The reference column is the main-table configuration, produced by the same entry point on the same annotation cache, so each sweep is a cell-for-cell comparison with exactly one changed knob.

G.1 Companion metrics and acquisition patterns

The other metric. Table 8 gives final accuracy for the main comparison, and Tables 9–11 give AUBC for the three ablations. All agree in sign with their counterparts in the main text, so no claim there rests on the choice of metric. The endpoint metric is also where the cost of reading a single operating point is clearest: QUORUM ranks second on final accuracy but last on AUBC, about 17 points below DLTA*.

Where each rule spends. Table 12 reports, for each acquisition rule, the fraction of instances whose votes all come from one model family — the quantity behind the discussion of concentrated spending in Section 4.3.

G.2 Sensitivity to modeling and deployment choices

What is and is not a tuning knob. Table 13 summarizes the six sweeps on one scale: the range each setting moves CALE's mean AUBC over its tested interval, divided by the reference cell's across-seed standard deviation. These ranges are sensitivity checks, not equivalence tests. The κ

Table 8: Main results, Final Accuracy (%) — the endpoint companion of Table 1. The compression this metric suffers at a generous budget is visible in the QUORUM row, competitive here and last by a wide margin on AUBC. Stars are as in Table 1; an unstarred cell marks a dataset on which the endpoint gap is not resolvable.

Method	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
Random	67.28±1.08***	67.23±0.59***	78.12±0.59***	61.45±0.87***	69.62±0.81***	68.74***
Cost-weighted Random	82.61±0.60***	77.72±0.61***	88.44±0.27***	73.87±0.48***	91.89±0.59***	82.90***
Cheap-First + DS	83.26±0.32***	77.92±0.58***	88.89±0.34***	74.45±0.61***	92.40±0.33***	83.39***
QUORUM	87.13±0.08***	80.38±0.07***	90.96±0.07***	**81.60**±0.06	**96.96**±0.05*	87.41***
LMCrowd	84.09±0.22***	82.18±0.17***	91.13±0.22***	77.76±0.30***	96.66±0.13***	86.36***
DLTA*	87.62±0.32	80.01±0.36***	90.96±1.37***	76.71±0.32***	96.91±0.44	86.44***
CALE (ours)	**87.97**±0.43	**84.03**±0.62	**91.96**±0.16	81.32±0.51	96.87±0.14	**88.43**

Table 9: Acquisition ablation, AUBC (%) — the anytime companion of Table 3.

Method	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
BALD / cost	78.01±0.29	73.24±0.45	84.51±0.37	68.34±0.42	84.38±0.37	77.70
Entropy / cost	69.03±1.49	51.78±1.08	68.83±2.01	67.83±0.96	84.77±0.32	68.45
AccEIG / cost	77.63±1.58	73.75±0.50	84.86±0.46	69.04±0.42	84.45±0.19	77.95
Proactive	77.93±0.48	73.81±0.54	84.57±0.61	68.91±0.54	84.35±0.22	77.92
CALE acquisition	**78.83**±0.24	**74.48**±0.26	**86.02**±0.65	**71.30**±0.22	**84.90**±0.12	**79.11**

Table 10: Aggregation ablation, AUBC (%) — the anytime companion of Table 4. All rows share one cost axis, so these differences are the aggregator's alone. The GLAD row is integrated on a coarser grid (Appendix F).

Aggregator	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
Majority vote	78.57±0.33	74.46±0.24	86.01±0.70	70.80±0.22	84.70±0.11	78.91
Dawid–Skene (K=1)	**79.00**±0.18	74.39±0.23	86.00±0.65	70.55±0.27	84.87±0.12	78.96
GLAD	78.88±0.19	74.16±0.16	**86.04**±0.67	70.90±0.20	84.82±0.12	78.96
CALE aggregator	78.83±0.24	**74.48**±0.26	86.02±0.65	**71.30**±0.22	**84.90**±0.12	**79.11**

Table 11: Regime ablation, AUBC (%) — the anytime companion of Table 2.

Variant	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
No regime	78.52±0.68	72.18±0.12	84.41±1.11	69.93±0.37	84.58±0.56	77.93
Regime only	**78.90**±0.06	72.25±0.22	83.89±2.52	67.28±2.44	84.26±0.97	77.32
Instance only	78.32±0.51	74.03±0.36	83.52±0.94	**72.14**±0.32	**85.38**±0.30	78.68
CALE	78.83±0.24	**74.48**±0.26	**86.02**±0.65	71.30±0.22	84.90±0.12	**79.11**

Table 12: Fraction of instances whose labels all come from a single model family, on the $B{=}4N$ traces. A single-family column identifies the confusion parameters only up to a label permutation, so a policy that concentrates its spending leaves the aggregator less to work with.

Acquisition rule	AG News	MedNLI	RTE	MMLU	DBPedia-14
BALD / cost	13.3%	18.8%	28.2%	25.0%	23.4%
Entropy / cost	64.4%	82.0%	89.4%	57.4%	82.9%
AccEIG / cost	17.5%	7.8%	9.0%	18.3%	11.4%
Proactive	28.7%	4.6%	25.1%	28.3%	12.1%
CALE acquisition	0.0%	1.5%	0.0%	6.5%	0.0%

sweep is the exception because its upper endpoint switches off the dependence model rather than merely changing a numerical setting.

Dirichlet concentration. Table 14 varies finite concentrations from $\kappa{=}0.01$ to 10 and also includes the $\kappa{\to}\infty$ approximation. Performance is flat between $\kappa{=}0.01$ and $\kappa{=}0.1$ (79.07 vs. 79.11) and then degrades to 76.68 in the independence limit: -2.43 AUBC and -4.32 Final Accuracy against the reference ($p{=}3.1\text{e}{-8}$ and $1.1\text{e}{-8}$). This limit removes the extra-multinomial dependence while

Table 13: How far each hyperparameter moves CALE's mean AUBC over its tested range, in units of the reference cell's pooled across-seed standard deviation (0.30 points). The ranges are descriptive sensitivity checks, not claims that tuning is unnecessary. The κ range includes the $\kappa \to \infty$ limit, where the dependence model is switched off altogether.

Hyperparameter	Range	Min	Max	Spread	Spread / seed sd
c_L/c_S	5–20	78.92	79.28	0.36	1.2×
Pool size	3S–3S+2L	78.60	79.11	0.50	1.7×
β	0.5–3	78.96	79.11	0.15	0.5×
K	2–5	79.01	79.12	0.11	0.4×
κ	0.01–∞	76.68	79.11	2.43	8.2×
R	10–60	79.06	79.13	0.07	0.2×

retaining the finite-K regime mixture; it is not ordinary $K{=}1$ Dawid–Skene. The gap is concentrated on MMLU (71.30 to 64.55), while DBPedia-14 barely moves (84.90 to 84.84). The near-flatness from $\kappa{=}0.01$ to 0.1 shows that the reference is not a finely tuned finite concentration.

Table 14: Sensitivity to the Dirichlet concentration κ at the reference setup of Section 4.1 with $K{=}3$, $\beta{=}2$. κ controls how far an instance-level confusion row may deviate from its regime-level mean; $\kappa{\to}\infty$ is the finite-K conditional-independence limit, approximated at 10^6.

Setting	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
$\kappa{=}0.01$	**78.90**	74.13	85.98	**71.41**	84.94	79.07
$\kappa{=}0.1^{\dagger}$	78.83	**74.48**	**86.02**	71.30	84.90	**79.11**
$\kappa{=}1$	78.62	73.83	85.86	70.23	**84.96**	78.70
$\kappa{=}10$	78.81	72.74	84.46	67.90	84.85	77.75
$\kappa{\to}\infty$	77.88	72.26	83.88	64.55	84.84	76.68

Number of latent regimes. Table 15 varies $K \in \{2, 3, 5\}$. The mean AUBC spans only 0.11 points; the tested choices are therefore practically stable around the fixed reference $K{=}3$. We do not extrapolate to much larger K, where finer partitions have fewer responses each and estimation noise should eventually dominate.

Table 15: Sensitivity to the number of latent regimes K, at the reference setup of Section 4.1 with $\kappa{=}0.1$, $\beta{=}2$.

Setting	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
$K{=}2$	**78.89**	**74.57**	85.70	**71.50**	**84.92**	**79.12**
$K{=}3^{\dagger}$	78.83	74.48	86.02	71.30	84.90	79.11
$K{=}5$	78.67	74.32	**86.05**	71.09	**84.92**	79.01

Exploration weight. Table 16 explores positive exploration weights $\beta \in \{0.5, 1, 2, 3\}$. The mean AUBC spans only 0.15 points across this range, so the global reliability-learning term is not sensitive to its exact positive weight in this setup.

Table 16: Exploration-weight robustness for CALE at the reference setup of Section 4.1 with $\kappa{=}0.1$, $K{=}3$. β weights the global reliability-learning term against the local label gain.

Setting	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
$\beta{=}0.5$	**79.01**	74.41	85.06	71.37	84.95	78.96
$\beta{=}1$	78.95	74.39	85.82	71.36	**84.96**	79.09
$\beta{=}2^{\dagger}$	78.83	**74.48**	86.02	71.30	84.90	**79.11**
$\beta{=}3$	78.88	74.28	**86.25**	71.23	84.89	**79.11**

EM refresh period. Table 17 varies $R \in \{10, 30, 60\}$, i.e. how many responses are acquired between empirical-Bayes refreshes. The mean AUBC spans 0.07 points — a fifth of a seed standard deviation — and neither $R{=}10$ nor $R{=}60$ differs significantly from $R{=}30$ ($p{=}0.90$ and $p{=}0.10$). Since only the refresh depends on R, the deployment reading is direct: it can be made six times less frequent, amortizing the EM cost accordingly, at no measurable accuracy cost.

Table 17: Sensitivity to the EM refresh period R at the reference setup of Section 4.1 with κ=0.1, K=3, β=2. R is the number of responses between empirical-Bayes refreshes—a computation–accuracy knob rather than a modelling choice.

Setting	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
R=10	78.86	**74.49**	**86.23**	71.23	84.85	**79.13**
R=30†	78.83	74.48	86.02	**71.30**	**84.90**	79.11
R=60	**78.88**	74.33	86.00	71.28	84.80	79.06

Cost ratio. Table 18 compares the four systems at $c_L/c_S \in \{5, 10, 15, 20\}$. CALE is best at every ratio and loses 0.36 AUBC across the whole $4\times$ range: a method that merely exploited a cheap large model would degrade sharply at 20:1. QUORUM is the informative contrast, dropping 10.56 (66.66 $\rightarrow$ 56.10), because its calibration phase front-loads a fixed number of large-model queries whose cost it does not adapt, whereas CALE's score falls back to inexpensive annotators exactly when the large model's marginal gain stops justifying its price.

Table 18: Cost-ratio robustness, AUBC (%); setup as in Section 4.1 except for the ratio. Only the large model's price moves (c_S=1 throughout), so a higher ratio buys fewer large labels within the same budget. † marks the reference ratio, which is the main-table configuration.

Method	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
$c_L/c_S = 5$						
QUORUM	64.70	65.06	77.01	61.51	65.00	66.66
LMCrowd	72.00	73.57	84.33	67.11	83.85	76.17
DLTA*	78.34	72.95	85.36	66.37	84.29	77.46
CALE	**79.30**	**74.42**	**86.01**	**71.73**	**84.93**	**79.28**
$c_L/c_S = 10^\dagger$						
QUORUM	58.00	59.39	72.69	54.72	55.80	60.12
LMCrowd	71.92	73.48	84.29	66.96	83.83	76.10
DLTA*	78.24	72.93	85.36	66.36	84.28	77.43
CALE	**78.83**	**74.48**	**86.02**	**71.30**	**84.90**	**79.11**
$c_L/c_S = 15$						
QUORUM	55.43	57.12	70.95	52.04	52.27	57.56
LMCrowd	71.91	73.44	84.28	66.91	83.82	76.07
DLTA*	78.24	72.92	85.36	66.36	84.27	77.43
CALE	**78.70**	**74.14**	**85.91**	**71.11**	**84.90**	**78.95**
$c_L/c_S = 20$						
QUORUM	53.80	55.98	70.06	50.58	50.08	56.10
LMCrowd	71.90	73.43	84.27	66.90	83.81	76.06
DLTA*	78.23	72.91	85.35	66.35	84.27	77.42
CALE	**78.72**	**74.09**	**85.86**	**71.03**	**84.90**	**78.92**

Model-pool size. Table 19 varies the pool between three small models (3S), the reference pool (3S+1L), and three small plus two large (3S+2L). CALE leads DLTA* on every pool, including the homogeneous one where no cost asymmetry exists to exploit. Removing the large model costs it 0.50 AUBC and 2.37 Final Accuracy.

Table 19: Model-pool-size robustness, AUBC (%); setup as in Section 4.1 except for the pool. LMCrowd and QUORUM require exactly one paid arm — LMCrowd one oracle, QUORUM one reference arm — so they are defined only on the reference pool and appear in that block alone.

Method	AG News	MedNLI	RTE	MMLU	DBPedia-14	Avg.
3S: three small models						
DLTA*	78.18	72.90	85.36	66.35	84.21	77.40
CALE	**78.34**	**73.46**	**85.51**	**70.78**	**84.92**	**78.60**
3S+1L: reference pool†						
QUORUM	58.00	59.39	72.69	54.72	55.80	60.12
LMCrowd	71.92	73.48	84.29	66.96	83.83	76.10
DLTA*	78.24	72.93	85.36	66.36	84.28	77.43
CALE	**78.83**	**74.48**	**86.02**	**71.30**	**84.90**	**79.11**
3S+2L: three small and two large						
DLTA*	78.28	72.93	85.39	66.36	84.29	77.45
CALE	**78.74**	**74.27**	**85.92**	**71.23**	**84.91**	**79.02**