Cost-aware LLM-based Online Dataset Annotation

Eray Can Elumar *
Dept. of Electrical and Computer Eng.
Carnegie Mellon University
Pittsburgh, PA 15213
eelumar@andrew.cmu.edu

Cem Tekin
Dept. of Electrical and Electronics Eng.
Bilkent University
Ankara, Türkiye
cemtekin@ee.bilkent.edu.tr

Osman Yağan
Dept. of Electrical and Computer Eng.
Carnegie Mellon University
Pittsburgh, PA 15213
oyagan@ece.cmu.edu

Abstract

Recent advances in large language models (LLMs) have enabled automated dataset labeling with minimal human supervision. While majority voting across multiple LLMs can improve label reliability by mitigating individual model biases, it incurs high computational costs due to repeated querying. In this work, we propose a novel online framework, Cost-aware Majority Voting (CaMVo), for efficient and accurate LLM-based dataset annotation. CaMVo adaptively selects a subset of LLMs for each data instance based on contextual embeddings, balancing confidence and cost without requiring pre-training or ground-truth labels. Leveraging a LinUCB-based selection mechanism and a Bayesian estimator over confidence scores, CaMVo estimates a lower bound on labeling accuracy for each LLM and aggregates responses through weighted majority voting. Our empirical evaluation on the MMLU and IMDB Movie Review datasets demonstrates that CaMVo achieves comparable or superior accuracy to full majority voting while significantly reducing labeling costs. This establishes CaMVo as a practical and robust solution for cost-efficient annotation in dynamic labeling environments.

1 Introduction

The rapid proliferation of data across domains has created an urgent need for accurate, large-scale annotation pipelines. While human experts and crowd workers have been the gold standard for dataset labeling, manual annotation is notoriously slow, expensive, and prone to inter-annotator inconsistency Petrović et al. [2020]. As machine learning models become increasingly sophisticated, their demand for high-quality, richly labeled datasets only intensifies, exacerbating this bottleneck.

Recent advances in large language models (LLMs) offer a promising remedy: by leveraging transformer-based architectures such as GPT, it is now possible to automate much of the labeling workload. LLMs excel at natural-language understanding, reasoning, and contextual inference, enabling rapid generation of annotations with minimal human effort Naveed et al. [2023].

However, relying on a single LLM introduces issues of biases inherited from its training data and stochastic variability across repeated queries, undermining reliability and reproducibility Errica et al. [2024], Li et al. [2024a]. A common strategy to bolster label quality is ensembling: querying multiple LLMs, or multiple samples from the same model, and aggregating their outputs via majority voting.

*Work done while at Carnegie Mellon University. Now at Intel Corporation.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

This reduces hallucinations and offsets the bias of individual models but substantially increases cost, as each additional model increases latency and compute expenditure Yang et al. [2023]. In practice, querying every available LLM for each instance is often wasteful and unnecessary.

In this paper, we address this trade-off by *adaptively selecting a subset* of LLMs for majority voting on each input, achieving comparable accuracy to full-ensemble voting while dramatically cutting cost. Unlike prior work on LLM weight optimization or query routing—which presumes access to ground-truth labels or a pre-trained routing model Chen et al. [2024], Nguyen et al. [2024], Ding et al. [2024]—our method operates *online*, without any held-out training set or ground truth.

Our contributions are as follows:

1. **Online formulation.** To the best of our knowledge, this is the first work on LLM-based dataset labeling in which both vote weights and the queried subset of LLMs are adapted in real time, i.e. without relying on a pre-trained model or a dedicated training set.
2. **Cost-aware Majority Voting (CaMVo).** We propose CaMVo, an algorithm that combines a LinUCB-style contextual bandit with a Bayesian Beta-mixture confidence estimator. For each candidate LLM, CaMVo computes a lower confidence bound on its correctness probability given the input's embedding, then selects the smallest-cost subset whose aggregated confidence exceeds a user-specified threshold δ.
3. **Empirical validation.** Through experiments on the MMLU benchmark and the IMDB Movie Review dataset, we show that CaMVo matches or exceeds the accuracy of full majority voting (majority voting with all available LLMs) while significantly reducing the cost: on MMLU, CaMVo achieves higher accuracy with around 40% lower cost; on IMDB, it attains only 0.17% drop in accuracy while halving query expenditure.

1.1 Related Work

Ensembling and Majority Voting with LLMs. Aggregating outputs from multiple LLMs (or repeated queries to a single LLM) via majority voting has become a popular strategy to boost annotation reliability. Chen et al. [2024] analyze the effect of repeated queries to a single model and observe a non-monotonic accuracy curve: performance improves initially but degrades beyond an optimal number of calls due to task heterogeneity. While additional LM calls enhance accuracy on easier queries, they may introduce noise or inconsistency that degrades performance on more challenging ones. To address this, the authors propose a scaling model that predicts the optimal number of LM calls required to maximize aggregate performance. Trad and Chehab [2024] compare re-querying and multi-model ensembles, showing that ensemble strategies are most effective when individual models or prompts exhibit comparable performance levels, and ensemble gains may diminish when individual model accuracies diverge.

Yang et al. [2023] propose a weighted majority-voting ensemble for medical QA, combining dynamic weight adjustment with clustering-based model selection. However, their approach relies on offline training data and focuses solely on improving accuracy, without accounting for the cost of querying. In contrast, our approach selects a cost-effective subset of heterogeneous LLMs online, without any pre-training.

LLM Query Routing. Query routing addresses the problem of selecting a single LLM per query to optimize cost or latency under performance constraints. Nguyen et al. [2024] cast LLM selection as a contextual bandit problem, training offline on labeled data to learn a routing policy that maps query embeddings to a single optimal LLM under a total budget constraint of b. Ding et al. [2024] train a router to distinguish "easy" versus "hard" queries, sending easy tasks to local LLMs and hard ones to cloud APIs. Unlike these methods, our method operates in a fully online setting without access to labeled training data, updating the model dynamically during the labeling process. Moreover, rather than routing to a single LLM, we select a cost-efficient subset for majority voting.

Confidence Estimation in LLM Outputs. Estimating model confidence can guide automatic label selection. Kadavath et al. [2022] explore how a language model's own uncertainty estimates can serve as predictors of answer correctness, and find that the cumulative log-probability the model assigns to its generated token sequence correlates strongly with factual accuracy across diverse benchmarks. Li et al. [2024a] generate code multiple times and use output similarity as a proxy for confidence,

choosing the most consistent result. While these works focus on self-consistency of a single model, we estimate a probabilistic lower bound on each LLM's correctness via a Bayesian Beta-mixture model, incorporating both past performance and context.

Weighted Majority Voting. Beyond simple voting, weighted schemes assign each annotator or model a reliability score such as the accuracy of the annotator, or the label confidence reported by the annotator. One notable approach is the GLAD model by Whitehill et al. [2009], which formulates weighted majority voting as a probabilistic inference problem over annotator expertise and task difficulty. GLAD jointly estimates per-annotator reliability and per-item ambiguity via a generative model, using an EM algorithm to infer the latent variables. Li and Yu [2014] introduce Iterative Weighted Majority Voting (IWMV) to aggregate noisy crowd labels by iteratively estimating worker reliability, and show it approaches the oracle Maximum A Posteriori (MAP) solution.

Crowdsourcing. Rangi and Franceschetti [2018] utilize the bandits-with-knapsacks framework to dynamically estimate worker accuracy and allocate tasks in real time to maximize overall labeling quality within a budget. Another influential model is by Raykar et al. [2010], which jointly infers true labels and annotator reliabilities by modeling each worker's confusion matrix. Through an expectation–maximization procedure, the method down-weights inconsistent annotators and yields more accurate aggregated labels without prior knowledge of worker quality. Our method parallels this online estimation but differs in that it leverages contextual embeddings and targets LLM ensembles rather than human annotators.

The comparison of our work with prior work is summarized in Table 1.

Method	Ensemble Type	Pretrained	Online	Contextual
Ours (CaMVo)	Subset voting	No	Yes	Yes
Yang et al. [2023]	Weighted voting	Yes	No	Yes
Nguyen et al. [2024]	Single-model routing	Yes	No	Yes
Ding et al. [2024]	Single-model routing	Yes	No	Yes
Chen et al. [2024]	Re-querying	No	No	No
Li et al. [2024a]	Re-querying	No	No	Yes
Li and Yu [2014]	Crowd aggregation	No	Yes	No
Rangi and Franceschetti [2018]	Crowd assignment	No	Yes	No
Raykar et al. [2010]	Crowd aggregation	No	Yes	No

Table 1: Comparison of our work with prior ensemble and routing approaches.

2 Problem Statement

In this section, we formally define the problem setting, introduce the baseline algorithm, and provide some results that will be used by our proposed algorithm, which will be introduced in §3.

Consider an unlabeled dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_T\}$, where each x_t denotes a data instance (e.g., a text sample). Let there be a set $[K]$ of K distinct large language models (LLMs), where each LLM l_i is associated with a known cost per token ρ_i and can be represented as a function $l_i : \mathcal{Q} \to \mathcal{R}$, mapping a query $q \in \mathcal{Q}$ to a response $r \in \mathcal{R}$. We denote the total number of possible labels for the dataset $\mathcal{D}$ as M. The objective in this setting is to assign a predicted label $\hat{y}_t \in [M]$ to each data instance x_t by querying a subset of the available LLMs and aggregating their outputs. Labeling is performed sequentially, where each data instance x_t is processed in round t. In each round, LLMs are queried independently of other LLMs and without memory of prior interactions (i.e., no context is preserved between rounds). Furthermore, all queries are made in a zero-shot setting, meaning that no task-specific fine-tuning or additional training data is used.

To serve as a baseline, we introduce the following weighted majority voting scheme. In this scheme, the predicted label $\hat{y}_t$ for instance x_t is determined by aggregating the votes of all K LLMs using:

$$\hat{y}_t = \arg\max_{m \in [M]} \sum_{i=1}^{K} \omega_{\text{def},i}(t) \cdot \mathbb{1}\{y_{i,t} = m\}, \tag{1}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function that returns 1 if the condition is true and 0 otherwise, and $y_{i,t}$ denotes the label outputted by LLM l_i for instance x_t. Since the true label is not available in our setting, we use the empirical accuracy of model l_i relative to the predicted labels as the voting weight. Hence, $\omega_{\text{def},i}(t) = (\sum_{s=1}^{t-1} \mathbb{I}(y_{i,s} = \hat{y}_s))/N_{i,t}$, where $N_{i,t}$ denotes the number of times LLM l_i has been queried up to round t, and $\hat{y}_s$ is the predicted label for round s. The pseudocode for this scheme, which we refer to as the *baseline method*, is provided in Algorithm 2 in Appendix C.

The goal is to label the dataset $\mathcal{D}$ in a cost-efficient manner by dynamically selecting a subset of LLMs for each data instance x_t. To facilitate this selection, we assume access to a model $\text{Emb}(\cdot)$ that generates a d-dimensional embedding $\mathbf{e}_t = \text{Emb}(x_t)$ for x_t in round t. This embedding serves as a representation of the instance and plays an analogous role to the *context* in a contextual bandit framework. Using this embedding, our approach, which will be introduced in § 3, estimates a lower bound on the probability that a given LLM l_i will produce the correct label for x_t, and utilizes this bound to determine the subset of LLMs. The details for estimating this lower bound is given in § 3.

Naturally, selecting only a subset of LLMs rather than querying all available models may result in reduced labeling accuracy. To manage this trade-off, we introduce a user-defined parameter $\delta \in [0, 1]$ that specifies the desired minimum relative confidence of the selected subset compared to the full majority vote using all K LLMs. Let $L_{i,t}$ denote the lower confidence bound on the estimated probability that LLM l_i correctly labels instance x_t at round t. Our algorithm identifies the cost-minimizing subset of LLMs whose aggregated confidence, relative to that of the baseline method, satisfies the accuracy constraint imposed by δ.

We will leverage the following result to estimate the confidence of the majority vote label based on the $L_{i,t}$ and $\omega_{i,t}$ values when majority voting is performed over a subset $\mathcal{A}$ of LLMs.

Lemma 2.1. *Let ω_i denote the weight and L_i the lower confidence bound on the correctness of the output from LLM l_i. Suppose the outputs of LLMs are conditionally independent given the data instance. Then, for a subset of LLMs $\mathcal{A} \subseteq [K]$, the lower bound on the probability that majority voting over the subset yields the correct label is given by*

$$\delta_{\mathcal{A}}(\boldsymbol{L}, \boldsymbol{\omega}) = \sum_{\substack{S \subseteq \mathcal{A} \\ \sum_{r \in S} \omega_r > \frac{W_{\mathcal{A}}}{2}}} \prod_{i \in S} L_i \prod_{j \in \mathcal{A} \setminus S} (1 - L_j), \tag{2}$$

where $W_{\mathcal{A}} = \sum_{i \in \mathcal{A}} \omega_i$. Proof of this result is provided in Appendix A.

For practical purposes and computational efficiency, this expression may be approximated as $\delta_{\mathcal{A}}(\boldsymbol{L}, \boldsymbol{\omega}) \approx 1 - F_{\text{Beta}}\left(0.5;\, W_{L,\mathcal{A}},\, W_{\mathcal{A}} - W_{L,\mathcal{A}}\right)$, where $F_{\text{Beta}}(x; \alpha, \beta)$ is the CDF of a $\text{Beta}(\alpha, \beta)$ distribution evaluated at point x, $W_{L,\mathcal{A}} = \sum_{i \in \mathcal{A}} \omega_i \cdot L_i$, and $W_{\mathcal{A}} = \sum_{i \in \mathcal{A}} \omega_i$.

Note that Lemma 2.1 can be extended to remove the conditional-independence assumption using Bahadur's model, which characterizes the joint distribution of binary variables using their marginal probabilities, second-order dependence terms known as Bahadur parameters, and higher-order interaction terms [Bahadur, 1961]. This framework allows the computation of majority vote confidence by summing the probabilities of all outcomes where the majority label is correct. However, the absence of a closed-form expression and the exponential growth in computational complexity with the number of variables limit its practical use. In practice, Monte Carlo–based simulation methods can approximate the majority-vote confidence by generating correlated LLM outputs; we describe and analyze CaMVo with one such approach in §G.

Finally, we introduce a user-specified parameter $k_{\min}$, which enforces a floor on the number of LLMs queried per instance. By requiring at least $k_{\min}$ votes in every round, this constraint further safeguards label quality—ensuring that no annotation is based on fewer than $k_{\min}$ model predictions.

The proposed Cost-aware Majority Voting (CaMVo) algorithm, which incorporates these results and user-defined parameters, is presented in §3.

3 The CaMVo Algorithm

We propose a novel algorithm, *Cost-aware Majority Voting* (CaMVo), for efficient dataset labeling with large language models (LLMs). CaMVo aims to select a cost-effective subset of LLMs for each input instance by leveraging contextual embeddings to estimate a lower confidence bound on

the probability that each model will produce a correct label. These bounds are computed using a LinUCB-based framework [Li et al., 2010]. Based on this information, CaMVo identifies a subset of LLMs such that the confidence of their weighted majority vote exceeds a user-specified threshold δ. If no such subset exists, the algorithm defaults to querying all available models. The pseudo-code of CaMVo is provided in Algorithm 1, and consists of six main steps described below.

Algorithm 1 Cost-aware Majority Voting (CaMVo) Algorithm

Input: Set of LLMs $[K]$, cost per token ρ_i for each LLM i, embedding model $\text{Emb}(\cdot)$, confidence threshold δ, LinUCB regularization parameter λ_L, exploration parameter α, regularization parameter λ_R
$A_{i,0} \leftarrow \lambda_L I_d,\ \boldsymbol{b}_{i,0} \leftarrow \mathbf{0}_d,\ \forall i \in [K]$
for each round $t = 1, 2, \ldots, T$ **do**
  Get context vector: $\mathbf{e}_t \leftarrow \text{Emb}(x_t)$
  for each LLM $i = 1, 2, \ldots, K$ **do**
    $q_{i,t}(\boldsymbol{e}_t) \leftarrow \mathbf{e}_t^\top A_{i,t-1}^{-1} \boldsymbol{b}_{i,t-1}$
    $\theta_{i,t}(\boldsymbol{e}_t) = q_{i,t}(\boldsymbol{e}_t) - \alpha\sqrt{\mathbf{e}_t^\top A_{i,t-1}^{-1}\mathbf{e}_t}$
    $\bar{L}_{i,t} \leftarrow \text{Est}_i(\theta_{i,t}(\boldsymbol{e}_t))$
    $L_{i,t} \leftarrow \frac{\bar{L}_{i,t}\cdot N_{i,t-1} + \lambda_R \cdot \log(t+1)/2}{N_{i,t-1} + \lambda_R \cdot \log(t+1)}$
    $\omega_{i,t} \leftarrow \mu_{i,t-1} \cdot q_{i,t}(\boldsymbol{e}_t)$
  end for
  $\mathcal{A}_t \leftarrow \text{Oracle}(\mathbf{L}_t, \boldsymbol{\omega}_t, \delta, k_{\min})$
  Query LLMs: $y_{i,t} = l_i(x_t),\ i \in \mathcal{A}_t$
  $\hat{y}_t \leftarrow \arg\max_m \sum_{i \in \mathcal{A}_t, y_{i,t}=m} \omega_i(t)$
  if $|\mathcal{A}_t| > 1$ **then**
    for each LLM $i \in \mathcal{A}_t$ **do**
      $r_{i,t} \leftarrow \mathbb{1}\{y_{i,t} = \hat{y}_t\}$
      Update: $N_{i,t} \leftarrow N_{i,t-1} + 1$
      Update: $A_{i,t} \leftarrow A_{i,t-1} + \mathbf{e}_t\mathbf{e}_t^\top$
      Update: $\boldsymbol{b}_{i,t} \leftarrow \boldsymbol{b}_{i,t-1} + r_{i,t}\mathbf{e}_t$
      Update: $\mu_{i,t} \leftarrow \frac{\sum_{s=1}^{t} \mathbb{1}\{y_{i,s} = \hat{y}_s\}}{N_{i,t}}$
      Update the parameters of Est_i
    end for
  end if
end for

LinUCB-Based Confidence Estimation. For each LLM l_i, CaMVo maintains a matrix $A_i \in \mathbb{R}^{d\times d}$ and vector $\boldsymbol{b}_i \in \mathbb{R}^d$, initialized as $A_{i,0} = \lambda_L I_d$, $\boldsymbol{b}_{i,0} = \mathbf{0}$, where $\lambda_L > 0$ is a user-defined parameter. Given $\boldsymbol{e}_t = \text{Emb}(x_t)$, the estimated confidence, and its confidence bound is computed as:

$$q_{i,t}(\boldsymbol{e}_t) = \boldsymbol{e}_t^\top A_{i,t-1}^{-1}\boldsymbol{b}_{i,t-1}, \quad C_{i,t}(\boldsymbol{e}_t) = \alpha\sqrt{\boldsymbol{e}_t^\top A_{i,t-1}^{-1}\boldsymbol{e}_t},$$

From these, the lower confidence bound (LCB) of LLM confidence can be found as:

$$\theta_{i,t}(\boldsymbol{e}_t) = q_{i,t}(\boldsymbol{e}_t) - C_{i,t}(\boldsymbol{e}_t).$$

Since this is an LCB of a probability, we clip it between 0 and 1 when it is out of range [0,1].

Bayesian Estimation of Label Correctness. Given the inherent probabilistic nature of LLM outputs, the estimated confidence score may not reliably indicate the correctness of a label. To address this, we introduce a Bayesian estimator $\text{Est}_i(\cdot)$ that models the posterior probability that l_i's prediction is correct, conditioned on this confidence. First, we define a latent variable $h_{i,t} = \mathbb{1}\{y_{i,t} = \hat{y}_t\}$, where $\hat{y}_t$ is the assigned label. Note that ideally, the true label should be used instead of $\hat{y}_t$, but since we do not have access to the true label, we instead use $\hat{y}_t$. We model the conditional likelihood of $q_{i,t}(\boldsymbol{e}_t)$, given the latent variable $h_{i,t}$, as a Beta-distributed random variable:

$$q_{i,t}(\boldsymbol{e}_t) \mid h_{i,t} = 1 \sim \text{Beta}(\alpha_{i,1}, \beta_{i,1}), \quad q_{i,t}(\boldsymbol{e}_t) \mid h_{i,t} = 0 \sim \text{Beta}(\alpha_{i,0}, \beta_{i,0}).$$

Further, to model $\mathbb{P}(h_{i,t} = 1)$, we use the empirical historical relative accuracy of l_i, $\mu_{i,t-1}$, which captures the accuracy of LLM l_i relative to the predicted labels up to round $t-1$. Similarly, $\mathbb{P}(h_{i,t} = 0) = 1 - \mu_{i,t-1}$. Applying Bayes' rule with these models, the posterior probability is modeled as:

$$\text{Est}_i(q) = \mathbb{P}(h_{i,t} = 1 \mid q) = \frac{\mu_{i,t-1} \cdot \text{Beta}_i(q; \alpha_{i,1}, \beta_{i,1})}{\mu_{i,t-1} \cdot \text{Beta}_i(q; \alpha_{i,1}, \beta_{i,1}) + (1 - \mu_{i,t-1}) \cdot \text{Beta}_i(q; \alpha_{i,0}, \beta_{i,0})},$$

We apply this estimator to $\theta_{i,t}(\boldsymbol{e}_t)$, the LCB of the estimated LLM confidence as $\bar{L}_{i,t} = \text{Est}_i(\theta_{i,t}(\boldsymbol{e}_t))$ to encourage exploration under the UCB principle. Note that unlike traditional UCB-based methods that promote exploration via upper bounds, we use the LCB as it expands the size of the set of LLMs likely to satisfy the confidence threshold δ.

Regularization. In the absence of ground-truth labels, empirical relative accuracy estimates in majority voting can overfit, resulting in overconfident weights and biased aggregation. Further, since subset selection is based on these estimated confidences, early estimation errors can bias the selection process and lead to compounding errors over time. To address this issue, we regularize the LCB of the estimated confidence of LLM l_i using Laplace smoothing:

$$L_{i,t} = \frac{\bar{L}_{i,t} \cdot N_{i,t} + \lambda_R \cdot \log(t+1)/2}{N_{i,t} + \lambda_R \cdot \log(t+1)} \tag{3}$$

where $\lambda_R > 0$ is a user-defined regularization parameter that controls the strength of smoothing. Laplace smoothing is chosen since it corresponds to using a uniform prior on the Beta estimator, and hence combines well with the Beta estimator. We use a $\log(t)$ term with the Laplace smoothing as compared to a constant term, as the $\log(t)$ term does not decay as quickly as the constant term and prevents overfitting in the long run.

Subset Selection with Oracle. We define the weight of LLM l_i for majority voting as $\omega_{i,t} := \mu_{i,t-1} \cdot q_{i,t}(\boldsymbol{e}_t)$. This way the weights of LLMs reflect both their past performance, and also their expected performance for the current data instance. An Oracle is used to find the lowest cost subset whose label confidence is above the threshold δ using the computed $L_{i,t}$ and $\omega_{i,t}$ values of LLMs. Using Lemma 2.1, this can be expressed as

$$\mathcal{A}_t = \text{Oracle}(\mathbf{L}_t, \boldsymbol{\omega}_t, \delta, k_{\min}) := \arg\min_{\mathcal{A}} c_{\mathcal{A}}(t) : \delta_{\mathcal{A}}(\mathbf{L}_t, \boldsymbol{\omega}_t) \geq \delta,\ |\mathcal{A}| \geq k_{\min} \tag{4}$$

where $c_{\mathcal{A}}(t) = \sum_{i \in \mathcal{A}} \rho_i \cdot H_i(x_t)$, ρ_i is the cost per token for LLM i, and $H_i(x_t)$ is the token count of the query to the LLM l_i for data instance x_t under the tokenization method of LLM l_i. Since we expect only one token as output (which will be the label for the data instance x_t), we ignore the output tokens. Note that as in Lemma 2.1, we assume that the outputs of LLMs are conditionally independent given the data instance. We analyze the correlated version of CaMVo that does not have this assumption in Appendix G. Also note that if no such $\mathcal{A}$ exists, CaMVo defaults to querying all LLMs.

Label Assignment. We query the LLMs in $\mathcal{A}_t$ and receive their responses. The label for x_t can be assigned via weighted majority vote using $\hat{y}_t = \arg\max_{m \in [M]} \sum_{i \in \mathcal{A}_t} \omega_i(t) \cdot \mathbb{1}\{y_{i,t} = m\}$.

Parameter Updates. If $|\mathcal{A}_t| > 1$, for each $l_i \in \mathcal{A}_t$, CaMVo updates $A_{i,t}$, $\boldsymbol{b}_{i,t}$, and $\mu_{i,t}$, which is the empirical relative mean accuracy, as:

$$A_{i,t} \leftarrow A_{i,t-1} + \boldsymbol{e}_t \boldsymbol{e}_t^\top, \quad \boldsymbol{b}_{i,t} \leftarrow \boldsymbol{b}_{i,t-1} + r_{i,t} \boldsymbol{e}_t, \quad \mu_{i,t} \leftarrow \frac{\sum_{s=1}^{t} \mathbb{1}\{y_{i,s} = \hat{y}_s\}}{N_{i,t}}$$

where $r_{i,t} = \mathbb{1}\{y_{i,t} = \hat{y}_t\}$. Note that parameters are not updated when $|\mathcal{A}_t| = 1$ as the reward will always be 1 in that case. The Beta distribution parameters $(\alpha_{i,h}, \beta_{i,h})$ can be updated via maximum likelihood estimation or a method-of-moments approximation based on the mean and variance of past confidence scores. These approaches are discussed in Appendix B.

4 Experiments

4.1 Experiments on the MMLU Dataset

We first evaluate CaMVo on the MMLU dataset Hendrycks et al. [2021a,b], a challenging multiple-choice benchmark spanning 57 diverse subjects including mathematics, U.S. history, law, and computer science. MMLU is well-suited to our setting, as it demands broad world knowledge and strong reasoning capabilities—conditions under which majority voting is particularly effective. To reduce computational cost, we restrict our evaluation to the test split, which contains 14,042 instances.

Models and setup. We use the following LLMs: Claude 3 Sonnet and Haiku from Anthropic Anthropic [2024], GPT-4o, o3-mini, and o1-mini from OpenAI OpenAI [2024], and LLaMA-3.3 and LLaMA-3.1 from Meta Meta [2024]. All models are queried using temperature 0.35 and top-$p = 1$, where applicable. To extract contextual embeddings for CaMVo, we use the 384-dimensional sentence transformer `all-MiniLM-L6-v2` Wang et al. [2020]. For computational efficiency, we approximate the confidence $\delta_{\mathcal{A}}(\boldsymbol{L}, \boldsymbol{\omega})$ using the cumulative distribution function of a Beta distribution. Further implementation details and experimental setup are provided in Appendix D.

Results. Table 2 (Left) reports the accuracy and cost of individual LLMs, as well as two baselines: *Majority Vote*, which aggregates all LLMs using weights proportional to their true accuracy, and the *Baseline Method*, which corresponds to Algorithm 2. *Cost* corresponds to the average input token cost when labeling the dataset and is reported in dollars per million input tokens. *Accuracy* reflects the percentage of data instances correctly labeled. CaMVo's performance under varying confidence thresholds δ and minimum vote counts $k_{\min} \in \{1, 3\}$ is shown in Table 3. Note that we include $k_{\min} = 3$ in our experiments as model parameters do not get updated when only a single LLM is queried, a scenario that can occur under $k_{\min} = 1$. As a result, users may prefer to avoid setting $k_{\min} = 1$. Moreover, for $k_{\min} = 2$, the selected pair of LLMs will always yield a majority vote in favor of the LLM with the higher weight, limiting the informativeness of the voting outcome. The algorithm is configured with $\alpha = 0.25$, $\lambda_R = 1$, and $\lambda_L = 1$. The *Target Accuracy* column reflects the targeted relative accuracy, which is the minimum relative accuracy CaMVo must exceed to satisfy the threshold δ, and is computed as $\delta \times$ (Majority Vote Accuracy) $= \delta \times 88.18\%$.

From Table 2, we observe that among individual models, o3-mini achieves the highest accuracy at 85.92%, while Majority Vote attains 88.18% at a substantially higher cost of $9.14 per million tokens. The Baseline Method also matches this accuracy and cost. Table 3 shows that CaMVo consistently meets or exceeds the desired accuracy levels specified by δ, across all settings of $k_{\min}$. From Table 3, it can be observed that CaMVo consistently satisfies all target accuracy levels defined by the confidence parameter δ. Moreover, the accuracy and cost of CaMVo exhibit a predictable trade-off: as δ decreases, the cost of labeling decreases accordingly, while accuracy also declines in a controlled manner. This behavior highlights the flexibility of CaMVo in adapting to a wide range of practical scenarios with varying accuracy and budget constraints. Further, at $\delta = 0.97$ and $k_{\min} = 1$, CaMVo achieves 88.33% accuracy at a cost of only $7.18, outperforming both baselines in cost-efficiency. This improvement stems from CaMVo's ability to dynamically select LLM subsets based on both global accuracy estimates and instance-specific contextual confidence.

We also observe that when $k_{\min} = 3$, lowering δ below 0.85 has negligible effect on cost or accuracy. This occurs because CaMVo settles on the lowest-cost trio: LLaMA-3.3, LLaMA-3.1, and Claude-3.5; whose combined cost ($1.44) represents a lower bound given the constraint on $k_{\min}$.

LLM / Method	Accuracy (%)	Cost
o3-mini	85.92	1.10
claude-3-7-sonnet	85.65	3.00
o1-mini	84.82	1.10
gpt-4o	83.58	2.50
llama-3.3-70b	81.70	0.59
llama-3.1-8b	68.01	0.05
claude-3-5-haiku	64.09	0.80
Majority Vote	88.18	9.14
Baseline Method	88.18	9.14

LLM / Method	Accuracy (%)	Cost
gpt-4o	95.68	2.50
o3-mini	95.40	1.10
claude-3-5-haiku	95.05	0.80
gpt-4o-mini	94.60	0.15
o1-mini	94.52	1.10
llama-3.1-8b	94.06	0.05
llama-3.3-70b	92.23	0.59
Majority Vote	95.62	6.29
Baseline Method	95.61	6.29

Table 2: Accuracy and cost of individual LLMs and baseline ensemble methods on the MMLU dataset (Left), and the IMDB Movie Reviews Dataset (Right).

Figure 1 (Left) illustrates the cost–accuracy trade-off of *All Subsets* versus CaMVo. Each gray point represents one of the $2^K - 1$ possible LLM subsets voted via ground-truth accuracies, while the yellow curve depicts those that are Pareto-optimal. CaMVo's results appear as blue markers for $k_{\min} = 1$ and green markers for $k_{\min} = 3$. The red marker denotes the Baseline Method. Remarkably, even without any a priori knowledge of LLM performance, or any pre-training, CaMVo consistently tracks, and sometimes surpasses the Pareto frontier, demonstrating its ability to approximate optimal cost–accuracy trade-offs in an online manner.

Figure 1 (Right) shows CaMVo's cumulative average accuracy (blue) and cost (red) for $\delta = 0.96$, $k_{\min} = 1$; the green horizontal line indicates the target accuracy. In early rounds, CaMVo explores larger, more expensive subsets, yielding both high cost and high accuracy. As the LCB estimates converge, the algorithm rapidly shifts to smaller, cheaper subsets that still satisfy the accuracy threshold. Cost declines steeply while accuracy stabilizes just above the target, illustrating CaMVo's ability to quickly identify and exploit the most cost-effective ensembles without sacrificing labeling quality. Additional plots with different parameters are provided in Appendix D.

CaMVo δ	Target Acc. (%)	Acc. (%) $k_{\min} = 1$	Cost $k_{\min} = 1$	Acc. (%) $k_{\min} = 3$	Cost $k_{\min} = 3$
0.99	87.30	88.47	9.14	88.47	9.14
0.98	86.42	88.59	8.57	88.59	8.57
0.975	85.98	88.49	7.80	88.49	7.80
0.97	85.53	88.35	6.67	88.33	6.67
0.965	85.09	88.27	5.66	88.27	5.66
0.96	84.65	87.98	4.74	88.03	4.74
0.955	84.21	87.40	3.38	87.01	3.36
0.95	83.77	86.82	2.76	87.01	2.96
0.90	79.36	84.88	1.19	84.80	1.81
0.85	74.95	84.41	1.03	82.14	1.58
0.80	70.54	82.12	0.70	81.32	1.51
0.75	66.14	68.80	0.16	81.24	1.50
0.70	61.73	68.38	0.14	81.22	1.50

Table 3: Accuracy and cost of CaMVo on the MMLU dataset under varying confidence thresholds δ and $k_{\min} \in \{1, 3\}$. For reference, the cost of the baseline method is \$9.14 per million tokens.

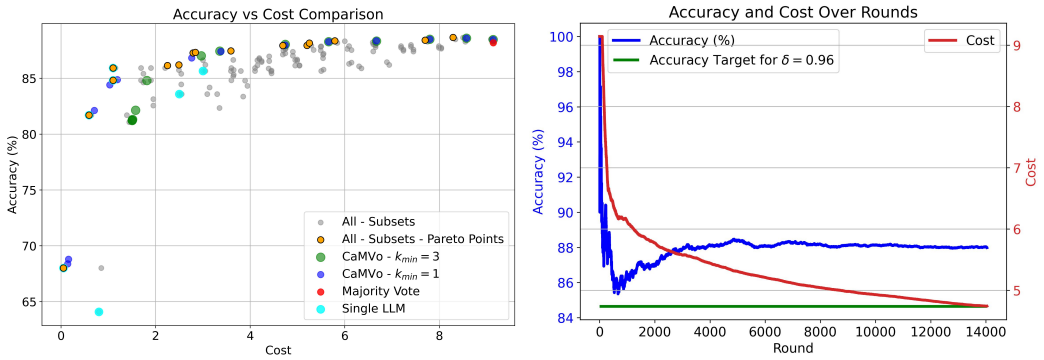

Figure 1: (Left) Cost–accuracy trade-off for MMLU dataset: gray dots show every LLM subset via weighted majority voting, yellow dots trace their Pareto-optimal frontier, blue markers are CaMVo at $k_{\min} = 1$, green markers at $k_{\min} = 3$, cyan markers denote the individual single LLMs, and the red marker denotes the Baseline Method. (Right) Cumulative average accuracy and cost of CaMVo with $\delta = 0.96$, $k_{\min} = 1$ over rounds.

4.2 Experiments on the IMDB Movie Reviews Dataset

We next test CaMVo on the IMDB Movie Reviews dataset Maas et al. [2011], a balanced binary-sentiment benchmark of 50,000 movie reviews. As before, we compare CaMVo against each individual LLM, a full-ensemble *Majority Vote*, and the *Baseline Method* (Algorithm 2).

Models and setup. We employ Anthropic's Claude 3-5 Haiku Anthropic [2024]; OpenAI's GPT-4o, o3-mini, GPT-4o-mini, and o1-mini OpenAI [2024]; and Meta's LLaMA-3.3 and LLaMA-3.1 Meta [2024]. All queries use temperature = 0.25 and top-p = 1, where applicable. We extract 384-dimensional contextual embeddings with `all-MiniLM-L6-v2` Wang et al. [2020] and approximate the confidence bound $\delta_{\mathcal{A}}(\boldsymbol{L}, \boldsymbol{\omega})$ via the Beta-CDF, as in §4.1.

Results. Table 2 (Right) reports the accuracy and cost (in dollars per million input tokens) of each LLM and the two baselines. The baseline underperforms the best individual model (95.68% vs. 95.61%) despite incurring a significantly higher cost. This is partly due to the relative ease of the IMDB Movie Reviews dataset, where individual LLMs already achieve high accuracy, limiting the marginal benefit of ensembling. As noted by Li et al. [2024b], ensemble gains are most pronounced on harder tasks. Additionally, Trad and Chehab [2024] highlight that large performance gaps among models can reduce ensemble effectiveness, making smaller, selective subsets preferable in such cases.

Table 4 presents CaMVo's accuracy–cost trade-off across various thresholds δ and $k_{\min} \in \{1, 3\}$. CaMVo's hyperparameters are $\alpha = 0.7$, $\lambda_R = 5$, and $\lambda_L = 1$; and the *Target Accuracy* is computed

similarly as $\delta \times 95.62\%$. Across all configurations, CaMVo meets or exceeds its target accuracy. Further, CaMVo achieves less than half the cost (when $\delta = 0.997$ and $k_{\min} = 1$) at a slightly lower accuracy of 95.45% compared to the baseline, confirming its practicality for large-scale sentiment annotation without any pre-training or ground-truth labels.

CaMVo δ	**Target Acc. (%)**	**Acc. (%) $k_{\min} = 1$**	**Cost $k_{\min} = 1$**	**Acc. (%) $k_{\min} = 3$**	**Cost $k_{\min} = 3$**
0.999	95.52	95.59	6.15	95.59	6.15
0.998	95.43	95.43	4.03	95.43	4.03
0.997	95.33	95.45	2.83	95.45	2.83
0.995	95.14	95.25	2.06	95.25	2.06
0.99	94.66	95.10	1.09	95.12	0.99
0.985	94.20	94.69	0.34	95.06	0.84
0.98	93.71	94.69	0.31	95.07	0.83
0.97	92.75	94.56	0.22	95.07	0.82
0.96	91.80	94.21	0.13	95.06	0.81
0.95	90.84	94.28	0.14	95.07	0.81
0.9	86.06	94.24	0.10	95.06	0.81

Table 4: Accuracy and cost of CaMVo on the IMDB dataset under varying confidence thresholds δ and $k_{\min} \in \{1, 3\}$. For reference, the cost of the baseline method is \$6.29 per million tokens.

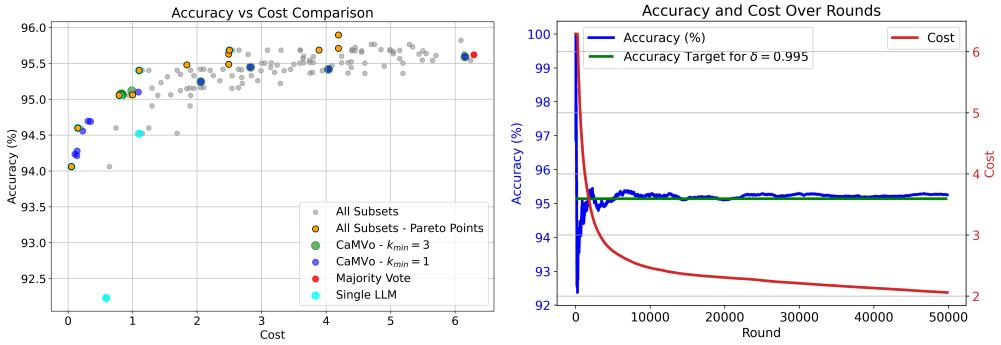

Figure 2: (Left) Cost–accuracy trade-off for IMDB dataset: gray dots show every LLM subset via weighted majority voting, yellow dots trace their Pareto-optimal frontier, blue markers are CaMVo at $k_{\min} = 1$, green markers at $k_{\min} = 3$, cyan markers denote the individual single LLMs, and the red marker denotes the Baseline Method. (Right) Empirical average accuracy and cost of CaMVo with $\delta = 0.995$, $k_{\min} = 1$ over rounds.

Figure 2 (Left) presents the analogous comparison of Figure 1 (Left) on the IMDB sentiment task. As before, gray points and the yellow Pareto-frontier points show all possible subset combinations, while blue and green markers plot CaMVo at $k_{\min} = 1$ and 3, respectively. The red marker denotes the Baseline Method. CaMVo closely matches the Pareto front in the low-cost regime (cost < 1), but lags behind in higher-cost regions. This exposes a key limitation: when majority voting with additional LLMs is ineffective, CaMVo's reliance on the independence assumption, which suggests that aggregating more LLMs improves accuracy; can lead to suboptimal performance.

Figure 2 (Right) plots CaMVo's cumulative average accuracy (blue) and cost (red) on IMDB with $\delta = 0.995$ and $k_{\min} = 1$; the green line marks the target accuracy. As before, early rounds involve querying larger, costlier ensembles to robustly explore each model's performance. Once the lower-confidence bounds stabilize, CaMVo swiftly transitions to minimal-cost subsets that still meet the accuracy requirement. This demonstrates CaMVo's rapid convergence to cost-effective model combinations without compromising annotation quality. Additional results for other parameter settings appear in Appendix E.

4.3 Additional Experiments

We present additional experimental results on the AG News Classification Dataset in Appendix F. Appendix G introduces CCaMVo (Correlated CaMVo), a practical extension of CaMVo that estimates subset confidence by modeling correlations among LLMs, thereby relaxing the independence assumption. We provide the corresponding algorithm, experiments, and a comparison with CaMVo, showing that accounting for correlations yields only marginal improvements in accuracy. Finally, Appendix H reports a sensitivity analysis evaluating how CaMVo's performance degrades under varying levels of correlation among LLM predictions.

5 Limitations

Our work relies on the assumption that the outputs of LLMs are independent of each other. Under this assumption, aggregating any subset of models with individual accuracy above 50% strictly improves majority-vote performance. In practice; i.e., on the IMDB sentiment task (§4.2), LLM outputs can be highly correlated, and majority voting may underperform the best single model. Consequently, CaMVo inherits these failures and can yield lower ensemble accuracy when independence is violated. Nevertheless, even in such regimes CaMVo still achieves the user-specified accuracy threshold while reducing cost relative to the full-ensemble baseline. This is mostly due to the fact that our results are relative to the full-ensemble baseline which also suffers from the same issue.

Appendix G introduces CCaMVo (Correlated CaMVo), a practical extension that estimates subset confidence thorugh estimating the correlation matrix among LLMs; this approach yields only marginal gains in accuracy. More broadly, extending CaMVo to account for inter-model correlations in a principled manner—e.g., via joint confidence estimation or diversity-aware subset selection—represents a promising and challenging direction for future work.

6 Conclusions

We have introduced Cost-aware Majority Voting (CaMVo), the first fully online framework for LLM-based dataset labeling that jointly adapts both vote weights and the subset of models queried on a per-instance basis. By combining a LinUCB-style contextual bandit with a Bayesian Beta-mixture confidence estimator, CaMVo estimates a lower bound on each LLM's correctness probability for the given input and selects the minimal-cost ensemble that meets a user-specified accuracy threshold.

Empirical results on the MMLU and IMDB benchmarks demonstrate that CaMVo matches or exceeds full-ensemble majority-vote accuracy while reducing labeling cost. On MMLU, CaMVo even surpasses the true Pareto frontier of all possible weighted subsets—despite having no prior knowledge of individual model performance. These findings establish CaMVo as a practical solution for cost-efficient, automated annotation in dynamic labeling environments without any ground-truth labels or offline training.

Our analysis assumes independence among LLM outputs, which can be violated in practice and may degrade ensemble gains. Nonetheless, CaMVo still enforces the user's accuracy target and delivers significant cost savings even under these conditions.

Our method can be naturally extended beyond classification tasks in several ways. For regression, LLM outputs can be aggregated via a weighted average or median, with weights updated based on how closely each LLM's output aligns with the aggregate. For ranking tasks, each LLM can assign scores to items, and a weighted combination of these scores would produce a final ranking; weights can then be adjusted according to agreement of the LLM's ranking generated from its scores with the aggregated ranking. These extensions involve only minor modifications to the aggregation and weight update steps, making them straightforward to integrate into our existing framework. Future work will explore diversity-aware selection and joint confidence models to mitigate correlated errors. We will also extend CaMVo to support iterative relabeling, allowing previously annotated instances to be revisited and refined as additional contextual information becomes available.

Acknowledgments and Disclosure of Funding

We thank Dr. Samarth Gupta for his thoughtful insights and stimulating discussions that helped shape the initial direction of this paper. This work was supported in part by the Office of Naval Research Grant # N00014-23-1-2275 and by CyLab Enterprise Security Initiative. The work of Cem Tekin was supported in part by the Scientific and Technological Research Council of Türkiye (TÜBİTAK) under Grant 124E065; by the Turkish Academy of Sciences Distinguished Young Scientist Award Program (TÜBA-GEBİP-2023); and by TÜBİTAK 2024 Incentive Award.

References

Anthropic. Claude 3 technical overview. https://www.anthropic.com/index/claude-3, 2024. Accessed: 2025-04-24.

R. R. Bahadur. On the classification based on responses to n dichotomous items. In Herbert Solomon, editor, *Studies in Item Analysis and Prediction*, pages 158–176. Stanford University Press, Stanford, CA, 1961.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei A Zaharia, and James Y Zou. Are more llm calls all you need? towards the scaling properties of compound ai systems. *Advances in Neural Information Processing Systems*, 37:45767–45790, 2024.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*, 2024.

Federico Errica, Giuseppe Siracusano, Davide Sanvito, and Roberto Bifulco. What did i do wrong? quantifying llms' sensitivity and consistency to prompt engineering. *arXiv preprint arXiv:2406.12334*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021b.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Hongwei Li and Bin Yu. Error rate bounds and iterative weighted majority voting for crowdsourcing. *arXiv preprint arXiv:1411.4086*, 2014.

Jia Li, Yuqi Zhu, Yongmin Li, Ge Li, and Zhi Jin. Showing llm-generated code selectively based on confidence of llms. *arXiv preprint arXiv:2410.03234*, 2024a.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need. *arXiv preprint arXiv:2402.05120*, 2024b.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web (WWW '10)*, pages 661–670. ACM, 2010.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Meta. Llama-3 models. https://www.llama.com/models/llama-3/, 2024. Accessed: 2025-04-24.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models. *arXiv preprint arXiv:2307.06435*, 2023.

Quang H Nguyen, Duy C Hoang, Juliette Decugis, Saurav Manchanda, Nitesh V Chawla, and Khoa D Doan. Metallm: A high-performant and cost-efficient dynamic framework for wrapping llms. *arXiv preprint arXiv:2407.10834*, 2024.

OpenAI. Openai models. https://platform.openai.com/docs/models, 2024. Accessed: 2025-04-24.

Nataša Petrović, Gabriel Moyà-Alcover, Javier Varona, and Antoni Jaume-i Capó. Crowdsourcing human-based computation for medical image analysis: A systematic literature review. *Health Informatics Journal*, 26(4):2446–2469, 2020.

Anshuka Rangi and Massimo Franceschetti. Multi-armed bandit algorithms for crowdsourcing systems with online estimation of workers' ability. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS 2018)*, pages 1345–1352, Stockholm, Sweden, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

Vikas C. Raykar, Shipeng Yu, Liangliang Zhao, Gilbert H. Valadez, Christopher Florin, Luca Bogoni, and Lawrence Moy. Learning from crowds. *Journal of Machine Learning Research*, 11(Apr): 1297–1322, 2010.

Fouad Trad and Ali Chehab. To ensemble or not: Assessing majority voting strategies for phishing detection with large language models. In *International Conference on Intelligent Systems and Pattern Recognition*, pages 158–173. Springer, 2024.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, pages 5776–5788. Curran Associates, Inc., 2020. URL https://arxiv.org/abs/2002.10957.

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. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22, pages 2035–2043, 2009.

Han Yang, Mingchen Li, Huixue Zhou, Yongkang Xiao, Qian Fang, and Rui Zhang. One llm is not enough: Harnessing the power of ensemble learning for medical question answering. *medRxiv*, 2023.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS) 28*, pages 649–657, 2015.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main scope of the paper is reducing costs while maintaining a user defined threshold δ on accuracy compared to majority voting with all LLMs, which is described in the Abstract and Introduction. In the abstract, we only claim performance gains compared to the full majority voting baseline on two datasets, which are MMLU and IMDB. These claims are backed by experimental results in §4. While we assume the responses of LLMs to be independent of each other in the design of our algorithm, we do not have this assumption in the experiments, and hence our claims do not depend on this assumption.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our work in the Limitations Section (§5).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate 'Limitations' section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We only have one minor theoretical result, which is Lemma 2.1. We have provided the independence assumption on the outputs of LLMs in the statement of the lemma, and we also provide the proof of the lemma in Appendix A. Note that this is a minor result to be used in the design of the algorithm.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have provided the pseudo-code of our algorithm, and the parameters used are listed in the Experiments section (§4). Additional details on the experimental setup, such as the prompt template for querying the LLMs, are available in Appendices D and E. The datasets used in the experiments are publicly available.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: While we do not directly provide data or code, our experimental results can easily be reproduced. We have provided the pseudo-code of our algorithm in Algorithm 1, and we have provided details on the experimental setting in our Experiments Section (§ 4), Appendices D and E. The datasets we used and the LLMs we queried are publicly available. Based on these, we believe our results can easily be reproduced.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We use online learning in our paper, and do not train any models beforehand. We have provided the hyperparameters used in each experiment in its relevant section. Datasets used in experiments are publicly available. We provide most of the important details regarding the experiments in our Experiments Section (§ 4); and we also provide additional details in Appendices D and E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Our experiments involve labeling a dataset with a given method, and for each method we record the average accuracy relative to ground truth labels, and the cost for the entire dataset. Since we have only one result per method, we do not report error bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer 'Yes' if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: LLMs were queried using online LLM providers. The experiments (apart from querying LLMs) do not consume much computer resources, and were performed with a personal computer.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics, and we confirm that our paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This paper considers optimizing the cost of LLM-based dataset labeling. Since we only introduce the optimization methods, and use only publicly available LLMs and datasets, we do not expect any societal impact of this paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No data or model is released with this paper, and the datasets used in our experiments are publicly available.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We provide the appropriate citations for the LLMs and the public datasets that we used in the paper in the Experiments Section (§4).

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Usage of LLMs is a core component of our paper as our paper studies cost optimization of LLM-based dataset labeling. In our Experiments Section (§ 4); and in Appendices D and E, we provide in detail how LLMs are queried to generate labels for the datasets.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.