Next Generation Active Learning: Mixture of LLMs in the Loop

Yuanyuan Qi[1], Xiaohao Yang[1], Jueqing Lu[1], Guoxiang Guo[1]
Joanne Enticott[1], Gang Liu[2*], Lan Du[1]

[1]Monash University
[2] Harbin Engineering University
{yuanyuan.qi, xiaohao.yang, jueqing.lu, guoxiang.guo, joanne.enticott, lan.du}@monash.edu, liugang@hrbeu.edu.cn

Abstract

With the rapid advancement and strong generalization capabilities of large language models (LLMs), they have been increasingly incorporated into the active learning pipelines as annotators to reduce annotation costs. However, considering the annotation quality, labels generated by LLMs often fall short of real-world applicability. To address this, we propose a novel active learning framework, Mixture of LLMs in the Loop Active Learning, replacing human annotators with labels generated through a Mixture-of-LLMs-based annotation model, aimed at enhancing LLM-based annotation robustness by aggregating the strengths of multiple LLMs. To further mitigate the impact of the noisy labels, we introduce annotation discrepancy and negative learning to identify the unreliable annotations and enhance learning effectiveness. Extensive experiments demonstrate that our framework achieves performance comparable to human annotation and consistently outperforms single-LLM baselines and other LLM-ensemble-based approaches. Moreover, our framework is built on lightweight LLMs, enabling it to operate fully on local machines in real-world applications.

Code — https://github.com/qijindou/MoLLIA
Appendix —
https://github.com/qijindou/MoLLIA/tree/main/Appx

Introduction

Active Learning (AL) is a paradigm in machine learning that strategically selects informative samples for annotation, with the objective of minimizing labeling costs while achieving predictive performance comparable to models trained on fully labeled datasets (Ren et al. 2021; Wu et al. 2025; Werner et al. 2024). With the rapid advancement of Large Language Models (LLMs) (Brown et al. 2020), renowned for their remarkable generalization capabilities (OpenAI 2023), these models have increasingly been integrated into the conventional active learning workflow, enhancing the cost-efficiency of the annotation process (Kholodna et al. 2024), and marking a significant evolution in the active learning landscape (Xia et al. 2025).

The integration of LLMs into the active learning workflow offers a promising route to improving annotation efficiency, however, their reliability as annotators remains an open challenge (Ding et al. 2023; Ming et al. 2024). Since LLMs are trained for general purpose tasks, their performance often degrades due to domain shift when applied to specialized datasets, resulting in annotation quality that is typically insufficient to serve as oracle labels (Gligorić et al. 2024; Guo et al. 2025). LLM-ensemble methods, which combine multiple LLMs with diverse architectures or training paradigms, offer a more robust and effective approach to enhancing annotation precision (Chen et al. 2025). Building upon this idea, leveraging the outputs from a Mixture of LLMs as annotators offers a feasible and more reliable strategy for transforming traditional human in the loop active learning into Mixture of LLMs in the loop active learning. Moreover, the active learning process can operate in a semi-fully, or even fully, human-free manner, while maintaining a relatively high standard of annotation quality.

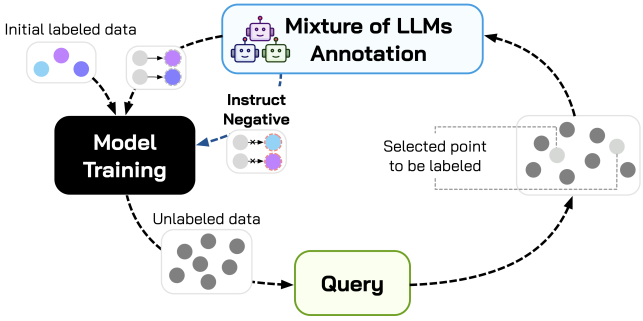

Figure 1: Overview of MoLLIA.

Although model performance typically scales with the volume of training data, real-world applications often suffer from a shortage of labeled data. In active learning settings, this limitation is even more pronounced, as usually only a small subset of ground truth labeled data is available at the initial iteration. Consequently, fully exploiting the limited labeled data to enhance the quality of Mixture of LLMs generated annotations is critical for the success of Mixture of LLMs in the loop active learning. Additionally, since the generated labels may be noisy or unreliable (Yang et al. 2025), using them directly as supervision labels can degrade model performance and compromise the reliability of the active learning process (Lu et al. 2025). To

*Corresponding author

mitigate this issue, adopting noisy label learning techniques that compensate for label noise has proven to be a promising solution for handling unreliable annotations (Song et al. 2022). Besides, regarding the AL model as a task specific small language model (SLM) which is trained or finetuned on the target dataset, the mismatch between the SLM and LLM-based annotators for annotation provides an additional insights of annotation discrepancy. And this discrepancy can be exploited to further refine the annotation process (Yuan et al. 2024). Integrating these strategies into the Mixture of LLMs framework allows the AL model to remain robust in the presence of imperfect annotations, thereby boosting both annotation quality and downstream model performance.

Overall, to further reduce human labeling costs while maintaining the reliability of active learning process for practical deployment, we propose the **M**ixture **o**f **LL**Ms **I**n the Loop **A**ctive Learning (MoLLIA) framework (Fig. 1). Specifically, at each active learning iteration, we select samples for annotation using existing acquisition strategies, and employ multiple lightweight LLMs to generate candidate labels. These outputs are then aggregated through a mixture module to determine the final annotation used in the next iteration training. To mitigate the effects of noisy labels, we further incorporate negative learning alongside annotation discrepancy between AL model and LLMs. In summary, our MLAL framework offers the following key contributions:

- **Human-Free Active Learning**: We propose a novel zero-human annotation active learning framework based on a Mixture-of-LLMs-based annotation model (MoLAM). By aggregating MoLAM as annotators and incorporating other learning mechanism, our method achieves annotation-free active learning with performance comparable to traditional human in the loop approaches.
- **Robust Active Learning**: We leverage the disagreement between the AL model (treated as a task specific SLM) and the LLM-based annotator as an annotation discrepancy indicator and incorporate a negative learning mechanism to improve the robustness of the learning process.
- **Reliable Empirical Validation**: We validate the effectiveness of proposed framework across four widely used benchmark datasets and multiple active learning strategies. MoLLIA achieves superior performance and demonstrates comparability to human annotators.

Related Work

With the rapid advancement of LLMs, their integration into active learning has become increasingly prevalent. Traditional AL methods rely on carefully designed uncertainty metrics and sample selection strategies to maximize model performance while minimizing annotation costs (Ren et al. 2021; Werner et al. 2024; Qi et al. 2025). Recently, due to their strong generalization capabilities and extensive inherent knowledge, LLMs have been incorporated into AL pipelines, either in the sampling or annotation stages, to further reduce labeling costs (Azeemi, Qazi, and Raza 2024; Xia et al. 2025). To utilize LLMs as annotators, Kholodna et al. (2024) employ inter-annotator agreement to evaluate the consistency of multiple LLMs and select the most reliable one to replace human annotators. Rouzegar and Makrehchi (2024) propose a hybrid annotation framework that combines LLM-generated labels with human annotations based on LLMs uncertainty. However, due to the limited quality of LLM-generated labels, these approaches either fail to match the performance of oracle labels or still require substantial human annotation effort. While methods such as NoisyAL (Yuan et al. 2024) and FreeAL(Xiao et al. 2023) incorporate both LLMs and smaller models to generate and refine labels, they are fundamentally noisy supervised learning approaches rather than active learning frameworks, as they lack iterative sample selection guided by trainable-model uncertainty. In addition, most existing methods depend on commercial API calls, raising unresolved concerns about data privacy and security, particularly in sensitive or real-world applications.

To ensure the reliability and effectiveness of LLM-generated outputs, numerous studies have explored techniques for estimating their quality, with a primary focus on uncertainty estimation. Overall, uncertainty estimation in LLMs can be broadly categorized into three main approaches: verbalization-based, consistency-based, and logit-based. Verbalization-based methods rely on prompting LLMs to self-assess its confidence by explicitly asking for likelihood judgments or uncertainty estimates through natural language responses (Yona, Aharoni, and Geva 2024; Lin, Hilton, and Evans 2022). Consistency-based methods estimate uncertainty by generating multiple responses for the same input and analyzing their variability (Chen and Mueller 2024; Tian et al. 2023). Logit-based methods derive uncertainty from the model's internal probability distribution, using metrics such as entropy or margin over predicted tokens to quantify confidence (Kuhn, Gal, and Farquhar 2023; Abbasi Yadkori et al. 2024; Zhang et al. 2025).

While uncertainty estimation offers valuable insights into the confidence of LLM predictions, it does not directly address the quality of the final annotations. To further enhance annotation reliability, LLM-ensemble methods have emerged as a widely adopted strategy that leverages the complementary strengths of multiple LLMs (Chen et al. 2025). Recent advances in this area can be broadly categorized into two groups: consensus-oriented and diversity-oriented approaches. Consensus-oriented methods aim to select the output that exhibits the highest agreement across multiple responses, often relying on voting or similarity metrics (Li et al. 2024a; Guha et al. 2024; Si et al. 2023). In contrast, diversity-oriented methods focus on analyzing of differences across candidate outputs to resolve conflicts or synthesize more informative and robust responses(Jiang, Ren, and Lin 2023; Tekin et al. 2024; Lv et al. 2024). However, given that label generation lacks semantic structure in the output space, consensus-oriented approaches are more suitable for our task, as they align better with the discrete and bounded nature of classification labels.

Methodology

Without loss of generality, let $L = \{X, Y\}$, $U = \{X\}$ represent the initial collection of training set and unlabeled

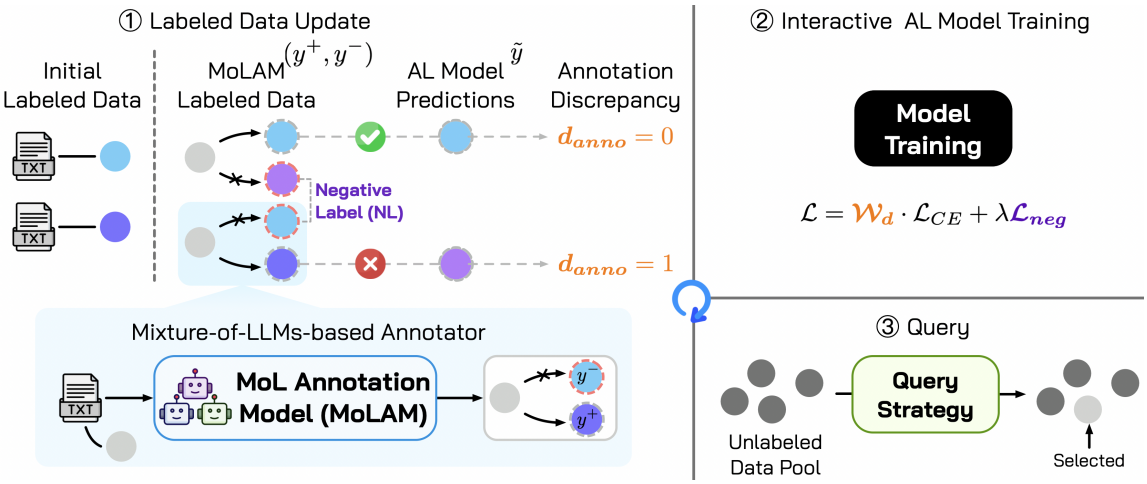

Figure 2: Workflow of MoLLIA framework. The AL model is first trained on the initial labeled dataset and used to query the most informative instances for annotation. A Mixture-of-LLMs-based annotator then generates labels y^+ and corresponding negative labels y^- for the selected instances (as detailed in Mixture-of-LLMs-based Annotation Model section). The annotation discrepancy (d_{anno}) is computed based on the disagreement between the AL model's predictions and the Mixture-of-LLMs-based annotator. The AL model is then updated using a loss function that incorporates both weighted annotation discrepancy and negative learning, and the querying process is iteratively repeated based on the updated AL model.

data samples, where $|U| \gg |L|$. Here, $Y \subset \{1, \ldots, K\}$ denotes the set of multi-class labels, and K is the total number of classes. Fig. 2 illustrates the workflow of our proposed framework, MoLLIA. Our method adopts standard active learning query strategies but replaces human annotation with the Mixture-of-LLMs-based Annotation Model (MoLAM). While MoLAM improves annotation quality compared to single LLM annotators, its outputs may still include noisy labels and therefore remain inferior to human level annotation quality. To further mitigate this effects, we propose Robust Active Learning, which enhances robustness through two key mechanisms. The first part utilize the negative labels, classes that an instance is unlikely to belong to, provided by MoLAM to guide the AL model away from incorrect predictions, thereby improving learning efficiency and class discrimination. The second part leverage the annotation discrepancy, quantifying the disagreement between MoLAM predicted labels and the AL model's predictions, to reweight the training loss, reducing the influence of potentially incorrect annotations.

Mixture-of-LLMs-based Annotation Model

Instead of relying on human annotators, MoLLIA further reduces annotation costs by introducing a fully human-free annotation model, MoLAM. The core idea of MoLAM is that LLMs with different architectures exhibit varying performance across datasets (Jiang, Ren, and Lin 2023). Therefore, by aggregating the outputs of several lightweight LLMs, MoLAM generates more reliable and comprehensive labels, delivering annotation quality that is acceptable for downstream active learning training.

Figure. 3 illustrates the training process of the MoLAM, including training data generation and model training. MoLAM is trained solely on the initial labeled dataset, which

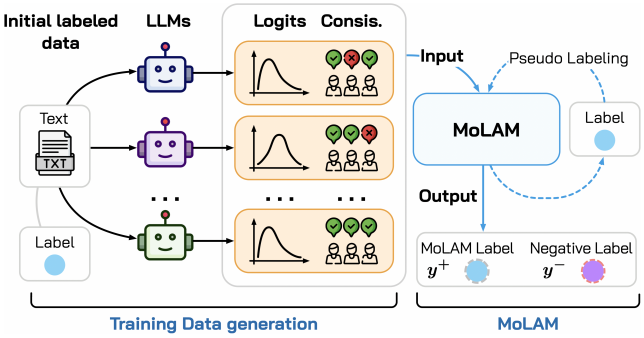

Figure 3: Overview of MoLAM.

comprises a very small portion of the entire data pool (only 50 instances). Let $\{x, y\}$ denotes one labeled example from the initial labeled set L, and let $\mathcal{M}_i \in \{\mathcal{M}_1, ..., \mathcal{M}_N\}$ represents the i_{th} LLM among the N LLMs involved in MoLAM. To construct the training data for MoLAM, each LLM $\mathcal{M}_i$ is queried T times on the same input x to produce: a logits vector, $z_i \in \mathbb{R}^K$, representing the model's confidence over K candidate labels; a consistency score, $c_i \in \mathbb{R}^K$, where each component $c_i^{(k)}$ indicates how frequently class $k \in \{1, \ldots, K\}$ is predicted across T generations. The computation is formalized as follows:

$$z_i = \mathcal{M}_i(x),\ z_i \in \mathbb{R}^K \tag{1}$$

$$\hat{y}_i^t = \text{Decode}(\mathcal{M}_i(x)),\ t = 1, \ldots, T \tag{2}$$

$$c_i^{(k)} = \frac{1}{T} \sum_{t=1}^{T} \mathbb{I}[\hat{y}_i^t = k],\ \forall k \in 1, \ldots, K \tag{3}$$

We then train MoLAM using the labeled data in the form of $\{[z_1, c_1, \ldots, z_N, c_N], y\}$, where each input consists of logits and consistency scores obtained from multiple LLMs.

Given the limited amount of labeled data available for supervision, we adopt a semi-supervised learning strategy based on a pseudo-labeling mechanism to leverage additional information from the unlabeled pool. A confidence threshold σ is applied to determine whether an unlabeled instance is reliable enough to be used for training. In this way, MoLAM encapsulates the collective knowledge of Mixture of LLMs and generates the refined label y^+, which are then used in subsequent active learning iterations. Additionally, to further exploit the expert knowledge implicitly encoded in the LLMs, MoLAM identifies negative labels y^-, defined as labels assigned consistently low probabilities (below threshold δ) by all LLMs. These negative labels are integrated into the training process via negative learning to improve the robustness and discriminative ability of the AL model. The overall procedure is formalized in the following equations:

$$\mathbf{h}(x) = [z_1, c_1, \ldots, z_N, c_N] \in \mathbb{R}^{2N \cdot K} \tag{4}$$

$$y^- = \{k | z_i^{(k)} < \delta,\ \forall i \in 1, \ldots, N\} \tag{5}$$

$$(y^+, y^-) = \text{MoLAM}(\mathbf{h}(x)) \tag{6}$$

Robust Active Learning

With MoLAM, the labels generated by a Mixture of LLMs become more reliable than single LLM. However, due to the inherent noise in LLM-generated annotations, we introduce robust active learning, which leverages implicit information embedded in both the LLMs outputs and the AL model predictions to further guide the training.

Specifically, we employ negative labels y^- as the set of classes that all LLMs assign consistently low confidence to, as defined in Eq. (5). To discourage the AL model from predicting these likely incorrect labels, we incorporate a negative learning loss $\mathcal{L}_{neg}$ into the training objective. This loss penalizes the model for assigning high probability to any class in y^-, and can be formulated as:

$$\mathcal{L}_{neg} = -\sum_{k \in y^-} \log(1 - p(k|x)) \tag{7}$$

where $p(k|x)$ is the predicted probability of class k by the AL model for input x, and $y^- \subset \{1, \ldots, K\}$ is the set of negative labels provided by MoLAM.

Moreover, the AL model can be regarded as task-specific SLM, trained for a particular task and thus more likely to encode domain-relevant knowledge. We apply the disagreement between the AL model predicted label and the MoLAM generated label as an indicator of annotation discrepancy, denoted as d_{anno}, and formularized as $d_{anno} = \mathbb{I}[\tilde{y} \neq y^+]$, where $\tilde{y} = \text{argmax}_k p(k|x)$, denotes the predicted class by the AL model. Empirically, we observe that d_{anno} is effective in identifying erroneous labels produced by LLMs. To avoid training leakage, we compute d_{anno} using the AL model's predictions from the previous iteration, before the newly selected samples have been incorporated into training. A detailed analysis of the effectiveness of d_{anno} is presented in the Component Effectiveness Analysis section under Experiments. We incorporate d_{anno} as a weight to emphasize high confidence annotations during training. We define the sample specific weight $\mathcal{W}_d$ as:

$$\mathcal{W}_d = \begin{cases} 1 & if \ \ d_{anno} = 0 \\ \alpha & if \ \ d_{anno} = 1 \end{cases} \tag{8}$$

where $\alpha \in (0, 1)$ is a down-weighting factor that reduces the influence of potentially incorrect annotations. The final loss function for the AL model is then defined as:

$$\mathcal{L} = \mathcal{W}_d \cdot \mathcal{L}_{CE} + \lambda \mathcal{L}_{neg} \tag{9}$$

where $\mathcal{L}_{CE}$ is the standard cross-entropy loss for multi-class classification, $\mathcal{L}_{neg}$ is the negative learning loss, and λ is hyperparameter controlling the weight of the negative learning penalty term. The complete training and update strategy of the MoLLIA framework is summarized in Algorithm 1.

Algorithm 1: MoLLIA Training and Update Strategy

Input: Labeled pool L; Unlabeled pool U; annotation model MoLAM; AL model; query size B; annotation discrepancy d_{anno}; negative labels y^-.
Output: Updated labeled and unlabeled pool, AL model, annotation discrepancy, negative labels.

for AL iteration **do**
  Select a batch of B instances $x = \{x_1, x_2, \ldots, x_B\}$ from U using the query strategy
  Obtain MoLAM-generated labels y^+ and negative labels y^- of x from MoLAM via Eq. (6)
  Update Labeled and unlabeled pool: $L \leftarrow L + \{(x, y^+)\}$; $U \leftarrow U - \{x\}$
  Obtain AL-predicted label $\tilde{y} = \arg\max_k p(k|x)$
  Compute negative learning loss $\mathcal{L}_{neg}$ via Eq. (7)
  Compute annotation discrepancy $d_{anno} = \mathbb{I}[\tilde{y} \neq y^+]$
  Compute weight $\mathcal{W}_d$ via Eq. (8)
  Update the AL model via Eq. (9)
end for

Experiments

To evaluate the performance and robustness of our proposed framework, we use four benchmark multi-class text classification datasets that are widely adopted in active learning research via Hugging Face platform[1]: AG News (Zhang, Zhao, and LeCun 2015), IMDB (Maas et al. 2011), TREC (Li and Roth 2002), and PubMed (Dernoncourt and Lee 2017). AG News is a news classification dataset composed of news titles and descriptions; IMDB is a collection of movie reviews for sentiment classification; TREC is a question classification dataset contains open-domain, fact-based questions; PubMed is a biomedical text classification dataset composed of article abstracts focused on diabetes-related topics. Table 1 provides a detailed summary of these datasets. The train, validation, and test splits used in our experiments follow the original dataset configurations.

Implementation

To balance both performance and deployability, we adopt five widely used lightweight LLMs, each ranging from

[1]https://huggingface.co/datasets

Dataset	#Vocab./ #Label	#Document		
		Train	Vali.	Test
AG News	65,043/4	114,000	6,000	7,600
IMDB	74,891/2	22,500	2,500	25,000
TREC	8,446/6	5,000	452	500
PubMed	45,457/5	176,642	29,672	29,578

Table 1: Experiment used dataset statistics.

7B to 9B parameters—suitable for inference on a single GPU with 24GB VRAM. The selected models include Gemma-2-9B-it, Llama-3.1-8B-Instruct, Mistral-7B-Instruct-v0.2, Qwen2.5-Coder-7B-Instruct, and Yi-1.5-9B. And the prompt is shown in Appendix.

We employed two pretrained language models as backbone classifiers, DistilBERT (Sanh 2019) and DistilRoBERTa (Liu et al. 2019), implemented using PyTorch (Paszke et al. 2019). To better simulate real-world deployment scenarios, we applied the cold start strategy (Zhu et al. 2019) with random initialization at the beginning of each active learning iteration (Frankle and Carbin 2018). All experiments were conducted on a single NVIDIA A40 GPU. The maximum input sequence length was set to 128 tokens, and each training iteration was run for up to 40 epochs. The initial labeled training set and query batch size were both set to 50 instances. To prevent overfitting and improve training efficiency, we applied early stopping with a patience of 10 epochs (Du et al. 2019; Ying 2019). We used the AdamW optimizer (Loshchilov and Hutter 2019), and the learning rate was set to 5e-5.

The annotation model, MoLAM, is implemented with XGBoost (Chen and Guestrin 2016). It is trained on 50 instances, identical to the initial labeled training set, randomly selected from the training set and validated on the corresponding validation set. The thresholds for pseudo-labeling and negative label identification are set to 0.9 and 0.001, respectively. The annotation discrepancy weight α fixed at 0.5, while the negative learning weight λ increases linearly from 0.4 to 1 during AL iteration. The XGBoost hyperparameter for each dataset are provided in Appendix.

Baselines

As the field of LLMs in the Loop active learning is still in its early stages and prior work primarily adopts either a single LLM as the annotator or relies on human–LLM hybrid setups, there is currently no established baseline for multi-LLMs annotation frameworks. Therefore, to provide a meaningful comparison, we evaluate our proposed MoLLIA framework against single-LLM annotation across four widely used active learning query strategies. To demonstrate the generalization capability of our framework, we adopt representative query strategies from three major categories: uncertainty-based, diversity-based, and hybrid approaches. NoiseStability (Li et al. 2024b), an uncertainty-based strategy, selects instances based on the variability of model predictions under perturbations. CoreSet (Sener and Savarese 2018), a diversity-based strategy, identifies a subset of samples that best represents the entire unlabeled pool by maximizing coverage in the feature space. BEMPS (Tan, Du, and Buntine 2023), a hybrid strategy, computes a proper scoring rule for each instance based on the model's predictive distribution, effectively capturing both uncertainty and representativeness. In addition, we include Random Sampling as a baseline to serve as a reference point for performance without active query selection.

	Methods	AG News	IMDB	TREC	PubMed
Single LLM	GEMMA	0.8349	0.9373	0.5741	0.7313
	LLAMA	0.7908	0.9172	0.4870	0.6233
	MISTRAL	0.8182	0.8835	0.6357	0.6257
	QWEN	0.7763	0.9403	0.6566	0.6217
	YI	0.7930	0.9503	0.7682	0.6755
LLM Ensem.	Vote-based	0.8247	0.9418	0.7320	0.6802
	Logits-based	0.8262	0.9421	0.7113	0.6802
Others	DA	0.5487	0.9206	0.4186	0.6864
	PAG	0.7771	0.9322	0.7152	0.7444
	SNAIL	0.8342	0.9489	0.6903	0.7199
	FixMatch	0.8530	0.9490	0.7207	0.7096
	MoL	0.8819	0.9534	0.7924	0.7744
	MoLAM	**0.8887**	**0.9538**	**0.8040**	**0.7772**

Table 2: Annotation accuracy across datasets for different approaches.

To evaluate the effectiveness of our annotation module, MoLAM, we compare it against a diverse set of baselines that share the common objective of enhancing annotation quality. These include approaches based on LLM-ensemble, data augmentation, meta-learning, and semi-supervised learning. Since the classification task does not involve semantically meaningful output text, we adopt two output-based LLM-ensemble methods as baselines: vote-based (Li et al. 2024a) and logits-based (Fathullah, Xia, and Gales 2023). These methods aggregate the predicted labels or the predicted probability distributions from multiple LLMs to produce the final annotation. To address the challenge of limited labeled data, we additionally consider vocabulary-level data augmentation (DA) (Ma 2019) and sentence paraphrasing (PAG) (Yadav, Tang, and Srinivasan 2024) as baselines, both aimed at enriching the input space. Furthermore, since the Mixture of LLMs paradigm can also be viewed as a form of black box meta-learning, we compare MoLAM with SNAIL (Mishra et al. 2018), a representative meta-learning method designed to adapt rapidly from few-shot examples. To assess the ability of MoLAM to utilize unlabeled data, we benchmark MoLAM against FixMatch (Sohn et al. 2020), a widely adopted semi-supervised learning method. Lastly, we include MoL, an ablation variant of MoLAM that excludes the pseudo-labeling mechanism, to isolate its contribution to overall performance.

Annotation Performance of MoLAM

Table. 2 presents the annotation performance of our proposed MoLAM compared with other annotation baselines, evaluated on a randomly sampled test set. For the first two

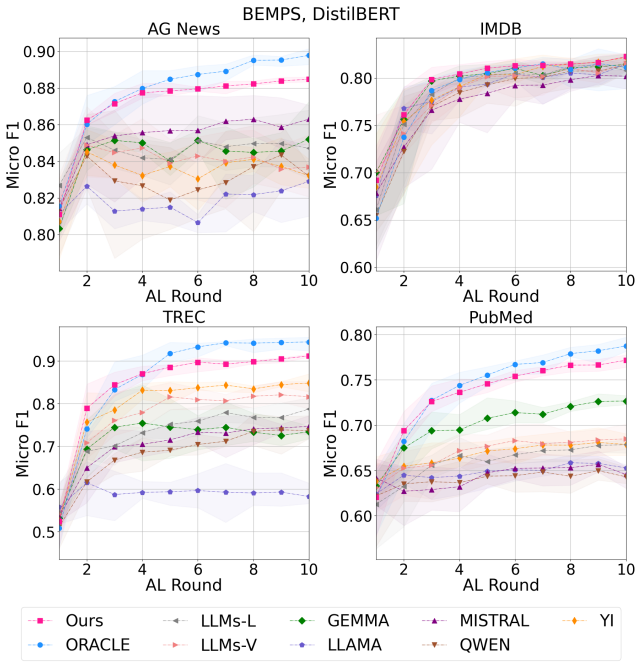

Figure 4: Averaged micro-F1 score with BEMPS on DistilBERT, averaged results with 5 random seeds.

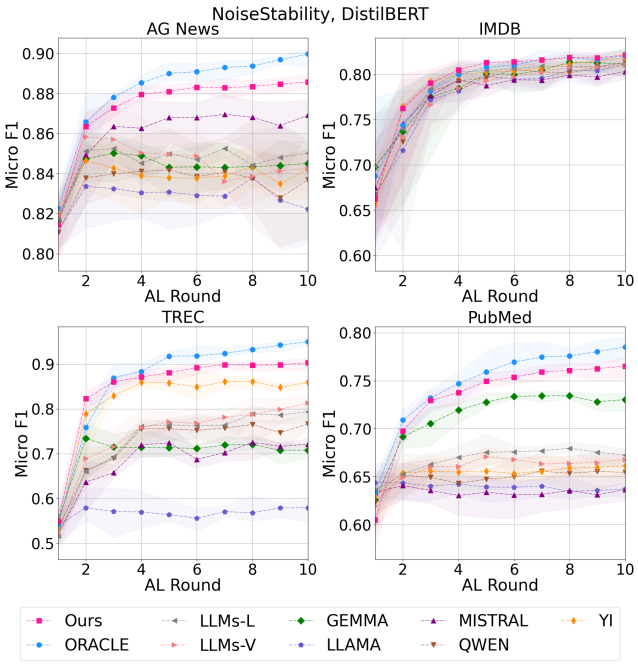

Figure 5: Averaged micro-F1 score with NoiseStability on DistilBERT, averaged results with 5 random seeds.

sections, Single LLM and LLM-ensemble, the reported accuracy is obtained by directly applying each method to label the input without any additional training. For the remaining methods involve training-based approaches, they are trained on 50 instances, same as the size of initial labeled pool. To maintain the consistency, the model backbones used for DA and PAG are XGBoost, the same as MoLAM. While for SNAIL and FixMatch, we adopt a MLP backbone with residual connections, in line with the architectural requirements of these methods, as XGBoost does not support gradient-based optimization.

From the table, we first observe that individual LLMs exhibit varied annotation performance across different datasets, and no single model consistently outperforms on all four benchmarks. This highlights the necessity of a Mixture-of-LLMs-based annotation model to improve generalization and robustness across diverse datasets. Additionally, we find that IMDB appears to be relatively easier for LLMs to annotate, whereas TREC and PubMed present greater challenges. This may be attributed to differences in the semantic meaning and complexity of the labels inherent to each dataset.

Among the methods that aggregate outputs from multiple LLMs, MoLAM consistently achieves the highest annotation performance across all datasets and baselines. While ensemble-based approaches help stabilize predictions by combining outputs from multiple LLMs, they do not lead to substantial improvement in annotation quality. Input data augmentation methods, DA and PAG, also exhibit limited effectiveness, likely because both vocabulary-level and sentence-level augmentations fail to enrich the input feature space; in particular, word-level transformations may distort the original semantics, resulting in degraded annotation performance. While advanced trainable methods, SNAIL and FixMatch, employ more complex architectures with attention mechanisms, their performance still falls short of MoLAM. This could be attributed to the limited expressiveness of the input features, which constrains the effectiveness of sophisticated architectures in this annotation setting. Moreover, the usage of pseudo-labeling in MoLAM plays a key role in effectively leveraging unlabeled data, further enhancing its annotation performance. We also present specific examples of labels generated by MoLAM compared with those from ensemble methods, as shown in Appendix Fig. 12.

Active Learning Performance

Fig. 4 and 5, along with Fig. 8 and 9 in the Appendix, illustrate the performance of our proposed framework across different datasets, backbone classifiers, and query strategies. LLMs-L and LLMs-V refer to the LLM-ensemble methods based on logits and voting, respectively. Overall, we observe that MoLLIA consistently outperforms both individual LLM and ensemble-based methods, achieving performance comparable to that of human annotators. Notably, on simpler datasets such as IMDB, which contains only two labels, MoLLIA even surpasses the oracle annotations. This is attributed to the incorporation of negative learning and annotation discrepancy, which enhance learning effectiveness. For datasets that are more challenging for LLMs to annotate, MoLLIA still demonstrates superior performance over both single LLM and ensemble methods, closely matches human-level annotation quality, and highlighting its potential as a practical substitute for human annotators. While

	AG News	IMDB	TREC	PubMed
True Negative Labels	0.4833	0.2142	0.2511	0.1844
False Negative Labels	0.0030	0.0002	0.0001	0.0022

Table 3: Distribution of negative labels across datasets, reported as the proportion relative to the full label space.

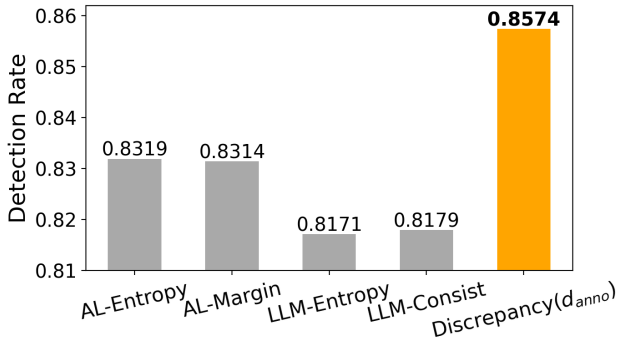

Figure 6: True positive detection rate of annotation discrepancy across different estimation methods, measured as the proportion of correctly identified accurate annotations among all annotated instances.

some single LLM, e.g. Llama and Yi, perform well on specific datasets, their performance is inconsistent and tends to drop significantly on others, highlighting the lack of generalization across diverse tasks.

Component Effective Analysis and Ablation Study

To evaluate the effectiveness of key components in our framework, we conduct a quantitative analysis of the negative labels and annotation discrepancy. Table 3 reports the distribution of negative labels y^- provided by MoLAM with the $\delta = 0.001$ across the total label space. True Negative Labels indicate cases where MoLAM correctly identifies labels that do not belong to the instance, while False Negative Labels correspond to instances where the true label is mistakenly classified as negative. The results demonstrate that MoLAM offers insightful guidance to the AL model through its negative label predictions. Importantly, the false negative rate remains extremely low across all datasets, suggesting that the use of negative labels introduces minimal additional noise into the learning process. Fig. 6 presents the true positive detection rate of annotation discrepancy, measured as the proportion of correctly identified accurate annotations out of all annotations. We compare our approach with several mainstream uncertainty estimation methods, including entropy, margin, and consistency-based metrics. The results show that the discrepancy derived from the disagreement between AL model and LLMs achieves the highest detection performance, highlighting its effectiveness in identifying potentially unreliable annotations.

To assess the contribution of each core component in our proposed MoLLIA framework, we conduct an ablation study with four variants, as summarized in Table 4. Each variant

	MoLAM	Negative Learning	Annotation Discrepancy
A	√	√	√
B	√	√	
C	√		√
D	√		

Table 4: Ablation study configurations.

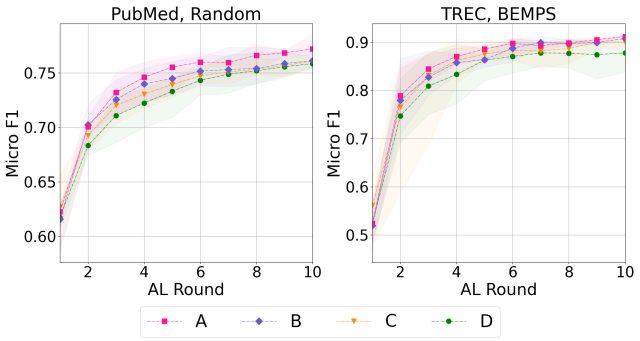

Figure 7: Ablation study of different component, and the legend is referred from Table 4.

disables one or more components, Negative Learning and Annotation Discrepancy, while retaining the MoLAM annotation module. The corresponding experimental results are presented in Fig. 7, with each curve labeled according to its configuration in the table. The results clearly show that both Negative Learning and Annotation Discrepancy significantly contribute to the overall performance of the model. Notably, the inclusion of Negative Learning yields the most substantial performance improvement, underscoring its critical role in enhancing the learning effectiveness of the AL model model by guiding it away from wrong predictions.

Additionally, we conduct a parameter sensitivity analysis to evaluate the generalizability of MoLLIA, as shown in Appendix Fig. 10 and Fig. 11. To demonstrate the deployability of the proposed framework, we report the maximum CUDA memory usage across different datasets and backbone models in Appendix Table 5.

Conclusion

In this study, we proposed a novel active learning framework that replaces human annotators with a Mixture-of-LLMs-based annotator, significantly reducing annotation costs. To ensure practical applicability and robustness in real-world scenarios, our framework relies solely on lightweight LLMs and incorporates negative learning and annotation discrepancy to further enhance the learning effectiveness of the AL model. Overall, the proposed MoLLIA framework demonstrates strong performance across four benchmark datasets, achieving annotation quality comparable to that of human annotators. Future work may explore hybrid annotation strategies that combine LLM-generated and human labels to balance efficiency and accuracy.

Acknowledgments

This research was supported by an Australian Government Research Training Program (RTP) Scholarship.

References

Abbasi Yadkori, Y.; Kuzborskij, I.; György, A.; and Szepesvari, C. 2024. To believe or not to believe your llm: Iterative prompting for estimating epistemic uncertainty. *Advances in Neural Information Processing Systems*, 37: 58077–58117.

Azeemi, A. H.; Qazi, I. A.; and Raza, A. A. 2024. Language Model-Driven Data Pruning Enables Efficient Active Learning. *arXiv preprint arXiv:2410.04275*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *34th Conference on Neural Information Processing Systems (NeurIPS)*, volume 33 of *Advances in Neural Information Processing Systems*.

Chen, J.; and Mueller, J. 2024. Quantifying Uncertainty in Answers from any Language Model and Enhancing their Trustworthiness. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5186–5200.

Chen, T.; and Guestrin, C. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

Chen, Z.; Li, J.; Chen, P.; Li, Z.; Sun, K.; Luo, Y.; Mao, Q.; Yang, D.; Sun, H.; and Yu, P. S. 2025. Harnessing multiple large language models: A survey on llm ensemble. *arXiv preprint arXiv:2502.18036*.

Dernoncourt, F.; and Lee, J. Y. 2017. PubMed 200k RCT: a Dataset for Sequential Sentence Classification in Medical Abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 308–313.

Ding, B.; Qin, C.; Liu, L.; Chia, Y. K.; Li, B.; Joty, S.; and Bing, L. 2023. Is GPT-3 a Good Data Annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 11173–11195.

Du, S.; Lee, J.; Li, H.; Wang, L.; and Zhai, X. 2019. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*. PMLR.

Fathullah, Y.; Xia, G.; and Gales, M. J. 2023. Logit-based ensemble distribution distillation for robust autoregressive sequence uncertainties. In *Uncertainty in Artificial Intelligence*, 582–591. PMLR.

Frankle, J.; and Carbin, M. 2018. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

Gligorić, K.; Zrnic, T.; Lee, C.; Candès, E. J.; and Jurafsky, D. 2024. Can Unconfident LLM Annotations Be Used for Confident Conclusions? *arXiv e-prints*, arXiv–2408.

Guha, N.; Chen, M.; Chow, T.; Khare, I.; and Re, C. 2024. Smoothie: Label free language model routing. *Advances in Neural Information Processing Systems*, 37.

Guo, G.; Aleti, A.; Neelofar, N.; Tantithamthavorn, C.; Qi, Y.; and Chen, T. Y. 2025. MORTAR: Metamorphic Multi-turn Testing for LLM-based Dialogue Systems. *arXiv preprint arXiv:2412.15557*.

Jiang, D.; Ren, X.; and Lin, B. Y. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14165–14178.

Kholodna, N.; Julka, S.; Khodadadi, M.; Gumus, M. N.; and Granitzer, M. 2024. Llms in the loop: Leveraging large language model annotations for active learning in low-resource languages. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 397–412. Springer.

Kuhn, L.; Gal, Y.; and Farquhar, S. 2023. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. In *The Eleventh International Conference on Learning Representations*.

Li, J.; Zhang, Q.; Yu, Y.; Fu, Q.; and Ye, D. 2024a. More agents is all you need. *Transactions on Machine Learning Research*.

Li, X.; and Roth, D. 2002. Learning Question Classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Li, X.; Yang, P.; Gu, Y.; Zhan, X.; Wang, T.; Xu, M.; and Xu, C. 2024b. Deep active learning with noise stability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 13655–13663.

Lin, S.; Hilton, J.; and Evans, O. 2022. Teaching Models to Express Their Uncertainty in Words. *Transactions on Machine Learning Research*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Lu, J.; Buntine, W.; Qi, Y.; Dipnall, J.; Gabbe, B.; and Du, L. 2025. Navigating Conflicting Views: Harnessing Trust for Learning. In *Forty-second International Conference on Machine Learning*. PMLR.

Lv, B.; Tang, C.; Zhang, Y.; Liu, X.; Luo, P.; and Yu, Y. 2024. URG: A Unified Ranking and Generation Method for Ensembling Language Models. In *Findings of the Association for Computational Linguistics ACL 2024*, 4421–4434.

Ma, E. 2019. NLP Augmentation. https://github.com/makcedward/nlpaug. Accessed: March, 2025.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of*

the Association for Computational Linguistics: Human Language Technologies, 142–150. Portland, Oregon, USA: Association for Computational Linguistics.

Ming, X.; Li, S.; Li, M.; He, L.; and Wang, Q. 2024. AutoLabel: Automated Textual Data Annotation Method Based on Active Learning and Large Language Model. In *International Conference on Knowledge Science, Engineering and Management*, 400–411. Springer.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2018. A Simple Neural Attentive Meta-Learner. In *International Conference on Learning Representations*.

OpenAI. 2023. GPT-4 Technical Report.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Qi, Y.; Lu, J.; Yang, X.; Enticott, J.; and Du, L. 2025. Multi-Label Bayesian Active Learning with Inter-Label Relationships. In *The 41st Conference on Uncertainty in Artificial Intelligence*.

Ren, P.; Xiao, Y.; Chang, X.; Huang, P.-Y.; Li, Z.; Gupta, B. B.; Chen, X.; and Wang, X. 2021. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9): 1–40.

Rouzegar, H.; and Makrehchi, M. 2024. Enhancing Text Classification through LLM-Driven Active Learning and Human Annotation. In *Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII)*, 98–111.

Sanh, V. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *Proceedings of 33rd Conference on Neural Information Processing Systems*.

Sener, O.; and Savarese, S. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. In *International Conference on Learning Representations*.

Si, C.; Shi, W.; Zhao, C.; Zettlemoyer, L.; and Boyd-Graber, J. 2023. Getting MoRE out of Mixture of Language Model Reasoning Experts. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 8234–8249.

Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C. A.; Cubuk, E. D.; Kurakin, A.; and Li, C.-L. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33: 596–608.

Song, H.; Kim, M.; Park, D.; Shin, Y.; and Lee, J.-G. 2022. Learning from noisy labels with deep neural networks: A survey. *IEEE transactions on neural networks and learning systems*, 34(11): 8135–8153.

Tan, W.; Du, L.; and Buntine, W. 2023. Bayesian estimate of mean proper scores for diversity-enhanced active learning. *IEEE transactions on pattern analysis and machine intelligence*, 46(5): 3463–3479.

Tekin, S.; Ilhan, F.; Huang, T.; Hu, S.; and Liu, L. 2024. LLM-TOPLA: Efficient LLM Ensemble by Maximising Diversity. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 11951–11966.

Tian, K.; Mitchell, E.; Zhou, A.; Sharma, A.; Rafailov, R.; Yao, H.; Finn, C.; and Manning, C. D. 2023. Just Ask for Calibration: Strategies for Eliciting Calibrated Confidence Scores from Language Models Fine-Tuned with Human Feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Werner, T.; Burchert, J.; Stubbemann, M.; and Schmidt-Thieme, L. 2024. A Cross-Domain Benchmark for Active Learning. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Wu, C.; Qi, Y.; Yang, X.; Lu, J.; Liu, G.; Buntine, W.; and Du, L. 2025. ALScope: A Unified Toolkit for Deep Active Learning. *arXiv preprint arXiv:2508.04937*.

Xia, Y.; Mukherjee, S.; Xie, Z.; Wu, J.; Li, X.; Aponte, R.; Lyu, H.; Barrow, J.; Chen, H.; Dernoncourt, F.; et al. 2025. From selection to generation: A survey of llm-based active learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14552–14569.

Xiao, R.; Dong, Y.; Zhao, J.; Wu, R.; Lin, M.; Chen, G.; and Wang, H. 2023. FreeAL: Towards Human-Free Active Learning in the Era of Large Language Models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 14520–14535.

Yadav, V.; Tang, Z.; and Srinivasan, V. 2024. Pag-llm: Paraphrase and aggregate with large language models for minimizing intent classification errors. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 2569–2573.

Yang, X.; Zhao, H.; Xu, W.; Qi, Y.; Lu, J.; Phung, D.; and Du, L. 2025. Neural topic modeling with large language models in the loop. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1377–1401.

Ying, X. 2019. An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, 022022. IOP Publishing.

Yona, G.; Aharoni, R.; and Geva, M. 2024. Can Large Language Models Faithfully Express Their Intrinsic Uncertainty in Words? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 7752–7764.

Yuan, B.; Chen, Y.; Zhang, Y.; and Jiang, W. 2024. Hide and Seek in Noise Labels: Noise-Robust Collaborative Active Learning with LLMs-Powered Assistance. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 10977–11011.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Zhang, Y.; Mehra, A.; Niu, S.; and Hamm, J. 2025. DPCore: Dynamic Prompt Coreset for Continual Test-Time Adaptation. In *Forty-second International Conference on Machine Learning*.

Zhu, Y.; Lin, J.; He, S.; Wang, B.; Guan, Z.; Liu, H.; and Cai, D. 2019. Addressing the item cold-start problem by attribute-driven active learning. *IEEE Transactions on Knowledge and Data Engineering*, 32(4): 631–644.

Appendix

Additional Result

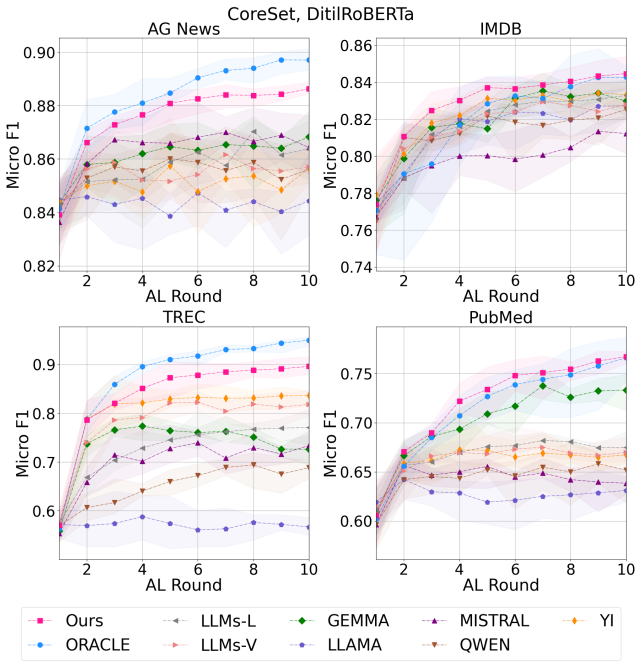

Figure 8: Averaged micro-F1 score with CoreSet on DistilRoBERTa, averaged results with 5 random seeds.

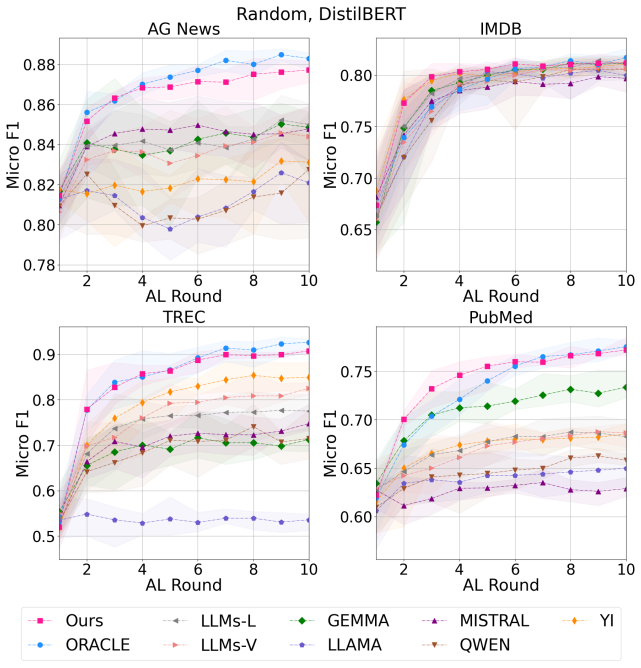

Figure 9: Averaged micro-F1 score with random on DistilBERT, averaged results with 5 random seeds.

Parameter Sensitive Analysis

Figures 10 and 11 present the parameter sensitivity analysis for the negative learning weight λ and the annotation discrepancy weight α. The results are reported as micro-F1 scores, averaged over five random seeds using the Random query strategy on DistilBERT. Since λ is linearly increased during AL iterations, the legends indicate its starting and ending values. Overall, the results show that increasing λ benefits the learning process by gradually incorporating negative labels. Additionally, setting $\alpha = 0.5$ allows the model to effectively leverage annotation discrepancies, helping it focus on more reliable supervision.

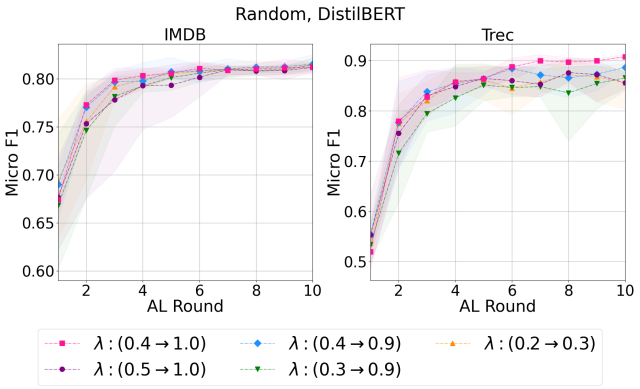

Figure 10: Performance on different negative learning weight parameter λ.

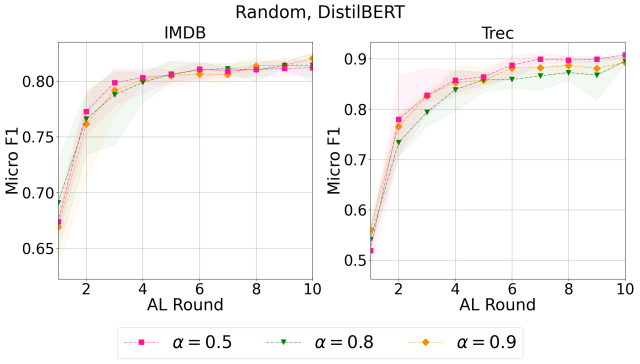

Figure 11: Performance with different values of the annotation discrepancy weight parameter α.

Qualitative Evaluation of MoLAM

Fig. 12 presents two examples illustrating the annotation performance of MoLAM. In these examples, LLMs-L and LLMs-V represent the LLM-ensemble baselines based on logits aggregation and majority voting, respectively. The values shown in square brackets correspond to the logits (for LLMs-L) or consistency scores (for LLMs-V) associated with each label index in the label list. In some cases, the sum of the consistency scores does not equal 1 because certain LLMs may generate non-existent or invalid labels.

```
"Label list":       ["negative", "positive"]
"Input text":       "excellent episode movie ala pulp fiction. 7 days -
                     7 suicides. it doesnt get more depressing than this.
                     movie rating : 8 \/ 10 music rating : 10 \/ 10"}
"GT label":         "positive"

"GEMMA-logits":     [0.9767777957,0.0229839049]
"GEMMA-consis.":    [1.0,0.0]
"LLAMA-logits":     [0.9156599855,0.085212484]
"LLAMA-consis.":    [1.0,0.0]
"MISTRAL-logits":   [0.9987564624,0.0000109721]
"MISTRAL-consis.":  [1.0,0.0]
"QWEN-logits":      [0.4611049891,0.5277478695]
"QWEN-consis.":     [0.5,0.5]
"YI-logits":        [0.0007171016,0.8430164922]
"YI-consis.":       [0.0,0.9]

"LLMs-L label":     "negative"
"LLMs-V label":     "negative"
"MoLAM label":      "positive"
```


(a) An example on IMDB.


```
"Label list":       ["world", "sports", "business", "science technology"]
"Input text":       "new mexico governor backs uc as los alamos lab manager
                     los angeles new mexico governor bill richardson today
                     endorsed the university of california to keep managing
                     the los alamos national laboratory."}
"GT label":         "science technology"

"GEMMA-logits":     [0.2373220176,0.0002307892,0.4434023127,0.3047787733]
"GEMMA-consis.":    [0.4,0.0,0.5,0.1]
"LLAMA-logits":     [0.2119691484,0.0002227128,0.5079409555,0.2716743574]
"LLAMA-consis.":    [0.1,0.0,0.7,0.2]
"MISTRAL-logits":   [0.0000087362,0.0000826602,0.0010006733,0.7540609537]
"MISTRAL-consis.":  [0.0,0.0,0.0,0.7]
"QWEN-logits":      [0.0050726533,0.0001707077,0.8523577452,0.1314268112]
"QWEN-consis.":     [0.0,0.0,1.0,0.0]
"YI-logits":        [0.0001573563,0.0000079763,0.000561323,0.9883978786]
"YI-consis.":       [0.0,0.0,0.0,1.0]

"LLMs-L label":     "science technology"
"LLMs-V label":     "business"
"MoLAM label":      "science technology"
```


(b) An example on AG News.

Figure 12: Qualitative comparison of annotation performance between LLM-ensemble and MoLAM.

All logits and consistency scores from the participating LLMs serve as input features to MoLAM. Notably, in both examples, MoLAM successfully predicts the correct label even when both ensemble baselines fail. This highlights the effectiveness of our proposed Mixture-of-LLMs-based annotation model in capturing nuanced decision patterns beyond simple aggregation, leading to more accurate and robust annotations.

Prompt design

> Classify the given question based on the following categories: **{List of labels}**
> Task: Determine the most appropriate category for the question. Your response should be only one of these labels: **{List of labels}**, with no additional text or explanation.
> Question: **{article}**
> Output:

MoLAM parameter

	AG News	IMDB	TREC	PubMed
Learning rate (lr)	0.07	0.01	0.05	0.01
Max depth (md)	5	5	6	3
# Estimators (ne)	300	300	300	500

Table 5: XGBoost hyperparameters used in MoLAM across datasets.

CUDA Memory Usage

	AG News	IMDB	TREC	PubMed
DistilBERT	20489	20780	20348	20114
DistilRoBERTa	20516	20604	19988	20340

Table 6: The maximum CUDA memory occupation (MB) during the AL iteration across different query strategies.