Unsupervised Mutual Learning among Comparable Large Language Models with Small Parameters

Jianjun Zeng[1], Weiyan Zhang[1], Yan Zhou[1], Chao Wang[2], Tong Ruan[1,*], and Jingping Liu[3,*]
[1]School of Information Science and Engineering,
East China University of Science and Technology, Shanghai, 200237, China
[2]Institute of Artificial Intelligence, Shanghai University, Shanghai, 200444, China
[3]School of Software Engineering, Sun Yat-sen University, Zhuhai, Guangdong, 519000, China
ruantong@ecust.edu.cn, liujp68@mail.sysu.edu.cn

***Abstract*—Recent progress in Large Language Models (LLMs) has increased interest in collaborative learning, typically grouped into two types: inference-based and training-based ones. However, both face challenges in a low-resource and small models (e.g., 7B-parameter LLMs) setting. The former contains instructions too complex for small models, while the latter require costly data. To this end, we propose Unsupervised Mutual Learning (UML), a novel framework that allows multiple small and comparable models to learn mutually using minimal unlabeled data. By leveraging inherent model and response diversity, UML establishes an iterative cycle where the models generate diverse outputs and learn from mutual input-output pairs. Experiments show that UML surpasses 8 baselines on 4 diverse tasks.**

***Index Terms*—Natural language processing, Large language model, Mutual learning, Unsupervised learning**

I. Introduction

Large Language Models (LLMs), particularly those with large parameters (referred to as large models), have proven successful in various tasks such as information extraction [1], question answering [2] and reasoning [3]. However, resource limitations remain a critical bottleneck for deploying large models in small and medium-sized enterprises. As a result, LLMs with small parameters (referred to as small models) provide a more practical alternative [4].

To address these challenges, collaborative learning, broadly classified as inference-based or training-based, has increased attention. The former aims to enable interactive collaboration during the inference phase without requiring additional training, such as CoDiscuss's inter-model discussion-augmented reasoning chain [5], COOP's cross-model knowledge gap probing [6], and ChatEval's role-diversified team debate [7]. However, when they are applied to small models, the limited instruction-following capability becomes a significant bottleneck leading to suboptimal collaboration. The latter facilitates knowledge transfer between different models during the training phase, using large-scale annotation, such as YODA's progressive fine-tuning [8], SRT interactive data refinement [9], and AutoAnno's automated annotations [10]. However, they require extensive and expensive annotation.

To address the challenges, we propose Unsupervised Mutual Learning (UML), a novel framework that enables multiple small and comparable models to mutually improve with minimal unlabeled data. Specifically, UML operates through an iterative cycle that begins with each model generating diverse outputs with the data as inputs. The outputs are then paired with the original inputs to construct a candidate set of input-output pairs. The pairs are selected and used as learning target to cross fine-tune the models, completing one round of UML. Further experiments reveal UML's underlying mechanisms: the models show effective unsupervised learning capability and response diversity plays a crucial role in model improvement. The code is at https://github.com/zengjianjun-ecust/UML.

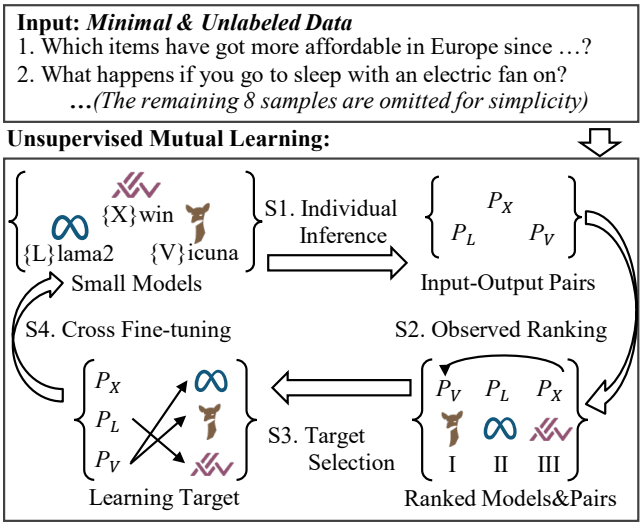

Fig. 1. The UML framework, illustrated using the TQA dataset.

We summarize the contributions of this paper as follows:

- We propose UML, a novel framework that enables small and comparable models to learn mutually through iterative cross fine-tuning with minimal unlabeled data.
- Recognizing and validating that the diversity derived from models is the key for UML, 5 selection strategies and 6 learning orders are proposed to exploit it systematically.
- Extensive experiments across 4 diverse tasks demonstrate that UML consistently outperforms all baseline methods.

*Corresponding authors: Tong Ruan and Jingping Liu.

TABLE I
PERFORMANCE OF LLMS ON 4 TASKS UNDER THE ZERO-SHOT SETTING. BOLD INDICATES THE HIGHEST SCORE PER TASK. ALL UNITS ARE %.

Model	TREX	TQA	CHold	TREC	AVG
Llama2	38.33	**36.33**	48.75	34.45	39.47
Vicuna	30.33	35.33	**50.97**	29.50	36.53
Xwin	**42.00**	33.67	37.81	**40.81**	38.57
AVG	36.89	35.11	45.84	34.92	38.19

II. TASK FORMULATION AND SOLUTION FRAMEWORK

In this section, we introduce the terminologies used in this paper, the problem formulation and our solution framework.

Term 1: **Model Comparability.** Given a set of tasks $\mathcal{T} = \{t_1, \ldots, t_t\}$ and models $\mathcal{M} = \{M_1, \ldots, M_m\}$ with scoring function $S(M_i, t_j)$, model comparability is defined by the normalized deviation from aggregate mean performance:

$$\mu = \frac{1}{mt} \sum_{M_i \in \mathcal{M}} \sum_{t_j \in \mathcal{T}} S(M_i, t_j), \tag{1}$$

$$\forall M_i \in \mathcal{M}, \left| \frac{1}{t} \sum_{t_j \in \mathcal{T}} S(M_i, t_j) - \mu \right| \leq \varepsilon, \tag{2}$$

where ε is a positive calculated margin ($\varepsilon = 1.66$ in Table I). The scoring function $S(M_i, t_j)$ computes the performance of model M_i on task t_j. The specific metrics used for each task are detailed in Section III-A.

Term 2: **Specialized Proficiency.** Despite model comparability, specialized proficiency is defined such that each model exhibits distinct specialized capabilities on at least one task. Formally, given $\mathcal{M}, \mathcal{T}$, the following condition must hold:

$$\begin{aligned} &\forall M_i, M_k \in \mathcal{M}, M_i \neq M_k, \exists t_j \in \mathcal{T}, \\ &S(M_i, t_j) > S(M_k, t_j) + \delta, \end{aligned} \tag{3}$$

with $\delta \geq 1.0$ indicating superiority. Table I demonstrates this phenomenon, where the models excel in different tasks.

Problem Formulation. Given a set of model $\mathcal{M}$ and unlabeled input X from task t_j, the problem is formulated as to find a function $f(X, \mathcal{M})$ that produces a new set of models $\mathcal{M}'$, which perform better on t_j, that is:

$$\begin{aligned} &\mathcal{M}' = f(X, \mathcal{M}) \\ &\forall i \in \{1, \ldots, m\}, S(M_i', t_j) > S(M_i, t_j). \end{aligned} \tag{4}$$

Solution Framework. To solve the above problem, we develop UML which leverages the response diversity to enable models to improve mutually. Specifically, as illustrated in Fig. 1, UML consists of 4 steps: ***Step 1: Individual Inference.*** To create input-output pairs, each model generates responses independently with a small set of unlabeled data as input. ***Step 2: Observed Ranking.*** To obtain the ranking information of models for Step 3, this step evaluates the quality of predictions from the previous step. Specifically, each model is evaluated on a small validation set using the scoring function, and models are ranked based on their scores. ***Step 3: Target Selection.*** For each model, the learning targets are chosen from the input-output pairs following a pre-defined selection strategy. ***Step 4: Cross Fine-tune.*** The models are cross fine-tuned on the data selected by the previous step. Step 1-4 repeat for multiple iterations until the predefined criteria are met.

Selection Strategies. To leverage the response diversity, we propose 5 selection strategies: ***Rotational Learning.*** A model learns from different models during successive iterations. For instance, if in the k-th iteration, model M_1 learned from M_2, it learns from M_3 in the $k+1$-th iteration. ***Random Ensemble.*** During each iteration, each model learns from input-output pairs where the outputs are randomly sampled from the pairs of other models. ***Global Ensemble.*** In each iteration, each model learns from input-output pairs where the pairs are randomly sampled from all models. ***Progress Boosting.*** According to the rank of the models after each iteration. If a model ranks higher, it continues learning from the same model. Otherwise, it switches to learn from the other model. ***Ranking Boosting.*** According to the rank of models, the model with the best performance (noted as M_b) learns from the second-weakest model (noted as M_s) and the weakest model (noted as M_w) in turn. Meanwhile, M_s and M_w always learning from M_b.

III. EXPERIMENT

A. Experimental Setup

Datasets and Evaluation Protocols. To assess broad capabilities, we select datasets from 4 task categories: cloze test TREX [11], open-ended QA TruthfulQA (TQA) [12], multiple-choice QA, CaseHOLD (CHold) [13] and multi-class classification TREC [14]. To construct the minimal and unlabeled dataset as input, we use 10 random training samples. We follow standard evaluation protocols: average accuracy for TREX [11], BLEU accuracy [15] for TQA, and F1-score for both CHold [13] and TREC [16]. For UML evaluation, we report both individual model performance and the average performance across all models in the ensemble.

Base Models. To meet the requirements for models with model comparability and specialized proficiency, we choose Vicuna-7B [17], Xwin-7B [18], and Llama2-7B [1] [19] as the base models. Table I presents the capabilities of these models on each tasks. Unless otherwise stated, we adopt the instruction-tuned version of model with 7B parameters. When investigating the mechanisms in UML, we introduce additional LLMs including Vigogne [20], Baichuan2 [21] as the 4-th and 5-th models. When investigating the model comparability, we introduce superior Llama3 [22] with 8B parameters. For fine-tuning and inference, We adopt the convention of the original authors of these models and follow the input-output template.

Baselines. We evaluate UML against a comprehensive set of 8 baselines, categorized into 2 types: inference-based and training-based ones. The former improves the performance by organizing multiple LLMs to leverage their diverse perspectives and capabilities through the designed workflows. This

[1]Llama3-8B does not meet the requirement for model comparability. Specifically, Llama3 is bigger in size and outperforms Llama2 by an average margin of 15%. Nonetheless, we provide experiments on Llama3.

TABLE II
MAIN RESULTS OF THE UML AND BASELINES.

Method	TREX	TQA	CHold	TREC	AVG
YODA	37.00	38.33	43.67	42.00	40.25
SRT	40.00	38.90	38.70	41.30	39.73
LbT	41.33	37.20	42.33	36.00	39.22
AutoAnno	35.67	38.40	43.67	40.00	39.44
CoDiscuss	38.00	37.00	46.33	40.66	40.50
ChatEval	37.33	36.67	49.33	38.33	40.42
COOP	38.03	36.67	45.33	41.33	40.34
CoLLM	41.00	37.33	49.00	40.66	42.00
UML(Ours)	**42.56**	**39.11**	**51.83**	**42.37**	**43.97**

includes 1) CoDiscuss [5], 2) ChatEval [7], 3) COOP [6], and 4) CoLLM [23]. The latter focus on leveraging an stronger LLM to guide and refine the fine-tuning process of the model to enhance its capabilities. The representative works are 1) YODA [8], 2) SRT [9], 3) LbT [24], and 4) AutoAnno [10]. All baselines are re-implemented using base models.

Implement Details. To efficiently fine-tune [25] the models, we adopt AdaLoRA [26] and follow [27] to apply the low-rank adaptation to the attention weight matrices in Transformer [28] block, specifically to q_proj, k_proj and v_proj. All experiments are conducted on a server running Ubuntu 20.04 LTS system, equipped with an Intel(R) Core(TM) i7-8700K CPU, 64 GB memory and a NVIDIA GeForce RTX 3090 24G GPU. One iteration for a model costs 2 training hours. Unless otherwise specified, we run UML for 4 iterations with the selection strategy of Ranking Boost.

B. Experimental Results

In this section, we conduct evaluation and systematic analysis of UML to investigate the following research questions: **Q1.** How does it compare to the baselines? **Q2.** What mechanism enables the improvement? **Q3.** What are the key factors?

Q1: Performance Gains. To answer Q1, we evaluate and compare UML against 8 baselines. From the Table II, we conclude that UML outperforms all baselines, specifically, leading the best-performing baseline, i.e, CoLLM, by 1.97%. The advantage of UML stems from addressing 2 limitations of the baselines. First, training-based baselines depend on a large amount of data and guidance from a stronger LLM, which is infeasible under our low-resource, small-model constraints. Second, although inference-based ones avoid fine-tuning, they suffer from poor coordination due to small models' limited capability to follow complex collaboration instruction. The UML solves both challenges through leveraging the model and response diversity in iterative cross fine-tuning. The result also reveals that the training-based baselines (rows 1-4) generally achieve lower scores compared to inference-based ones (rows 5-8), due to their reliance large models, while in inference-based ones, small models retain basic task-solving abilities.

Q2: Mechanisms. To address Q2, we conduct experiments by varying model count and the results lead to the conclusion that: 1) LLMs are capable of unsupervised learning, and 2) LLMs can benefit from the diverse responses. To substantiate

TABLE III
PERFORMANCE OF UML WITH VARYING MODEL COUNTS. THE BOLD DENOTES THE HIGHEST SCORE ACROSS VARYING MODEL COUNTS.

Task	The Count of Models in UML				
	1	2	3	4	5
TREX	42.78	40.78	42.56	44.11	45.44
TQA	38.33	39.11	39.11	42.22	42.33
CHold	50.62	51.17	51.83	47.60	45.75
TREC	36.94	37.69	42.37	41.36	39.96
AVG	42.17	42.19	**43.97**	43.82	43.37

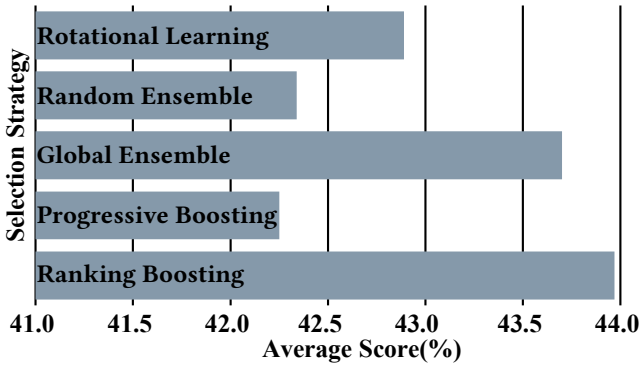

Fig. 2. Comparing various selection strategies. The strategies are listed on bar along the horizontal axis.

the first point, we compare the scores in Table I with the single model (referred to as self-learning) in Table III, finding that self-learning (i.e., the model learning from its own responses) yields an average performance gain of 3.98%. This demonstrates the same underlying principle as [29] that the models can benefit from unlabeled data with random labels. To verify the second conclusion, we increase model diversity by incorporating more models to UML, which increases response diversity. From Table III, we observe that as the count increased from 1 to 3, the average performance improves continuously. This reinforces a shared underlying principle that the models can benefit from diversity (whether via multiple models or diverse templates [30]). As the count increases to 5, the performance of tasks diverges: generation ones, i.e., TREX and TQA, benefit from increased diversity, while classification ones, i.e., CHold and TREC, suffer from the noise that causes deviation from the labeling schema.

Q3: Analysis of Selection Strategies. To address Q3, we propose and evaluate 5 selection strategies for performing the Step 2 Target Selection. In Fig. 2, the best results are achieved by the Ranking Boosting strategy. The strategy involves the weaker models learning primarily from the strongest, while the strongest model alternates between learning from weaker and second-strongest models. The former allows weaker models to directly benefit from the stronger model, resulting in immediate improvement, while the latter enables the strongest model to learn from a diverse set of responses, thereby optimizing its output behavior for the specific task.

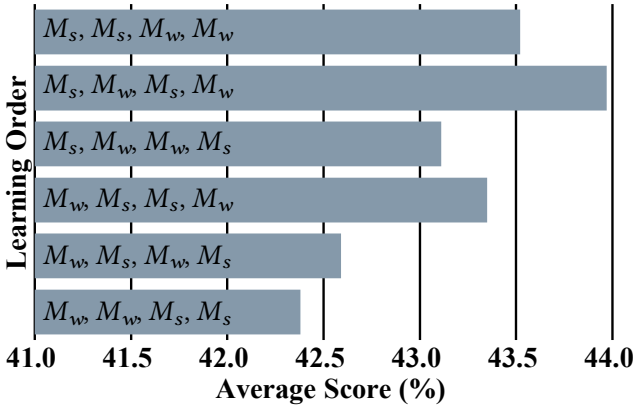

Fig. 3. Comparing various learning orders while using the ranking boosting strategy. The orders are listed on bar along the horizontal axis. The order in which M_b is learning from is shown along the y-axis. M_b, M_s, M_w denotes the best, the second weakest and the weakest models.

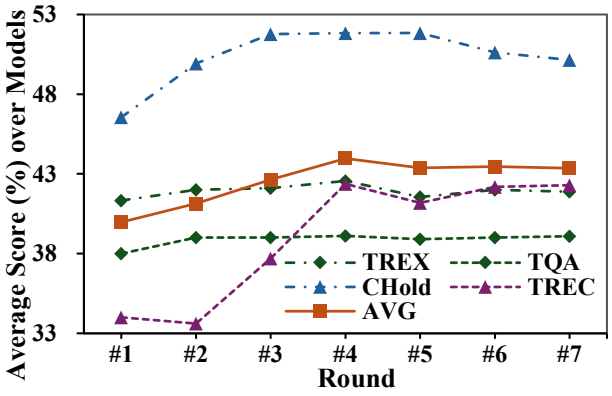

Fig. 4. The performance of UML by extending UML to 7 iterations. The dashed lines correspond to the performance of different tasks. The red line with squares denotes the average over four tasks.

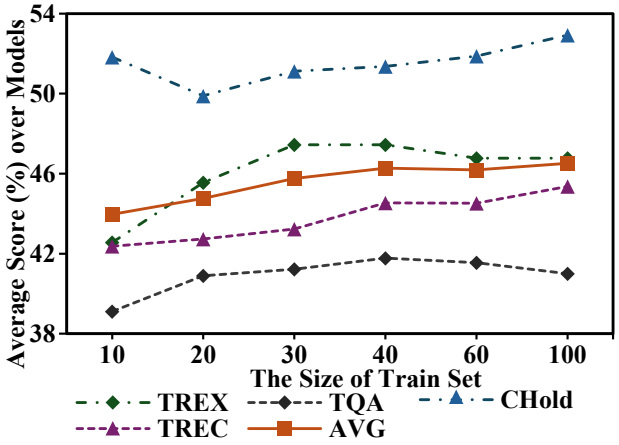

Fig. 5. The performance comparison with varying train set sizes in UML.

Q3: Analysis of Learning Orders. To address Q3, we compare various learning orders to identify the optimal learning order for the optimal selection strategy ranking boosting. Specifically, we compare 6 different learning orders. The goal is to determine a learning order that maximizes the benefits of UML while preserving diversity in their responses. See Fig. 3 for the detailed learning order. As shown in Fig. 3, the order where M_b learns from M_s, M_w, M_s, and M_w yields the best performance. This is because the first round of learning from the second-weakest model strikes a balance between learning from stronger models with incorporating diversity.

Q3: Analysis of Iteration in UML. To address Q3, we explore the effect of increasing the number of iteration rounds. As shown in Fig. 4, optimal performance is achieved at the 4th iteration, with limited improvement observed in subsequent rounds. This is because the system contains 3 models and by the 4th round, M_b has already learned from the other models twice, and the diversity of responses has been fully learned, reaching a point of diminishing returns.

Q3: Analysis of Train Set Size. To investigate Q3, we quantify the performance scaling of UML across a range of unlabeled training set sizes (10 to 100 samples). Fig. 5 shows that UML performance on all tasks improves with training set size from 10 to 40 samples, peaking at 40 samples. This improvement stems from increased response diversity enabled by additional data within the UML framework. However, as the train set size grows from 40 to 100 samples, the performance of classification tasks, i,e., CHold and TREC, continues to improve, where as that of generation tasks, i.e., TQA and TREX, declines. This trend reverses when the count of models increases from 3 to 5 in Table III. The reversal occurs because with 40 samples, the response diversity from the models has already been fully leveraged. More data does not enhance diversity but instead shifts the model's focus toward task-specific patterns, aligning it more closely with the task's inherent schema. This shift benefits classification tasks, which have a well-defined solution space, i.e., a fixed labeling schema, allowing them to further improve. While the focus on task-specific patterns correspondingly reduces the model's effectiveness in open-ended, infinite solution-space scenarios, resulting the performance decline in generation tasks.

Q3: Analysis of Diversity. To Analyze the relationship between diversity and iterative improvement, we employ Semantic Entropy (SE) and Discrete Semantic Entropy (DSE) [31] to quantify response diversity. In this experiment, we track entropy and task accuracy on TREX across iterations from the lens of Llama2, denoted as L#0–L#7. As shown in Fig. 6, both entropy and accuracy are initially low at L#0. After the first UMA iteration (L#1), we observe a substantial increase in semantic entropy. This surge in diversity arises from the introduction of mutual alignment of heterogeneous models. Notably, this rise in entropy coincides with a marked improvement accuracy. The concurrent enhancement of both metrics suggests that within greater semantic diversity is associated with better performance.

Q3: Analysis of Model Comparability. To investigate Q3, we relax the constraint by introducing a superior model that dominates all other LLMs. We define the default models, i.e,

TABLE IV
PERFORMANCE OF ML-COMPARABLE AND ML-SINGULAR FROM INDIVIDUAL MODEL AND OVERALL PERSPECTIVE.

Perspective	Method	TREX	TQA	CHold	TREC	AVG
Vicuna	Zero-Shot	30.33	35.33	50.97	29.50	36.53
	Self-Learning	40.67	35.67	56.82	33.50	41.66
	ML-Comparable	39.33	40.67	59.17	39.21	44.60
	ML-Singular	39.33	37.33	59.98	53.74	47.60
Xwin	Zero-Shot	42.00	33.67	37.81	40.81	38.57
	Self-Learning	48.00	41.33	43.94	38.60	42.97
	ML-Comparable	47.00	37.00	44.63	51.32	44.99
	ML-Singular	46.67	37.33	52.19	47.27	45.87
Llama2	Zero-Shot	38.33	36.33	48.75	34.45	39.47
	Self-Learning	39.67	38.00	51.09	38.71	41.87
	ML-Comparable	41.33	39.67	51.68	36.56	42.31
Llama3	Zero-Shot	55.67	43.33	59.57	51.95	52.63
	Self-Learning	69.52	42.33	57.32	64.20	58.34
	ML-Singular	53.33	43.00	57.02	57.34	52.67
Overall Gain	ML-Comparable	5.67	4.00	5.99	7.44	5.77
	ML-Singular	3.78	1.78	6.95	12.03	6.13

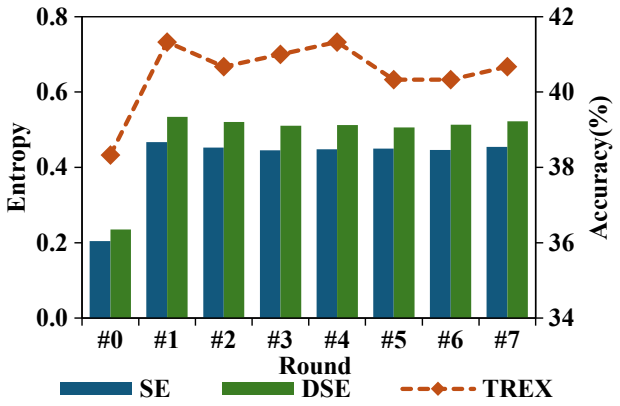

Fig. 6. The evolution of diversity and accuracy of Llama2 on the TREX task across rounds. The left y-axis shows entropy, with vertical bars representing Semantic Entropy (SE) and Discrete Semantic Entropy (DSE); the right y-axis displays the accuracy on TREX, depicted as a polyline.

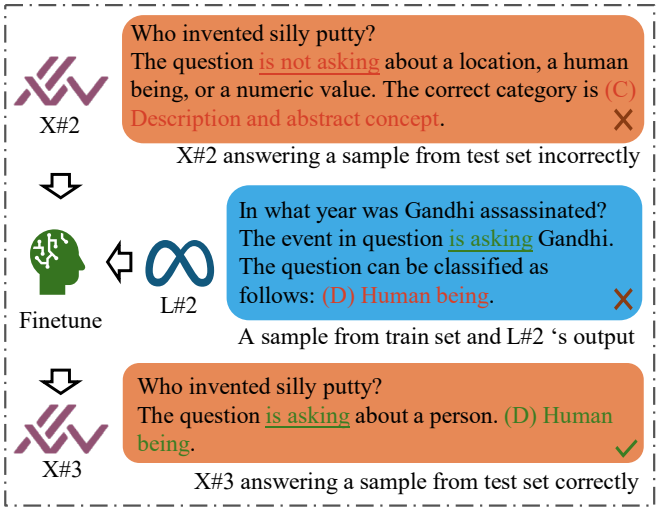

Fig. 7. Case Study: Xwin's update from iteration 2 to 3 in UML.

Llama2, Vicuna and Xwin, as ML-Comparable. By replacing Llama2 with the superior Llama3, we create ML-Singular. We then evaluate both configurations against zero-shot and self-learning setting. In Table IV, ML-Comparable achieves overall gains of 5.77%, while ML-Singular demonstrates a slightly higher improvement of 6.13%. Notably, the performance gains of ML-Comparable are close to that achieved by learning directly from ML-Singular. Crucially, UML achieves competitive effectiveness without relying on a stronger model, by fully leveraging the diversity among comparable models. Although no performance gains are observed for Llama3 when compared to the zero-shot baseline in ML-Singular, the overall improvement remains significant. This improvement can be attributed to a distillation-like effect, where weaker models benefit by learning from Llama3.

Case Study. In Fig. 7, we introduce a case study of UML from the perspective of Xwin, where X#2 is refined into X#3 (the notation M#n indicates model M after the n-th iteration). To identify the learning target under the ranking boost strategy, we first determine model ranking based on the Step 2 Observed Ranking: X#2 is the strongest model, Vicuna#2 (V#2) the weakest, and Llama2#2 (L#2) the second weakest. Applying the optimal learning order as shown in Fig. 3, the derivation is that X#2 is refined into X#3 by learning from L#2. In Fig. 7, it can be observed that X#2 provides an answer by rejecting other options, resulting in an incorrect answer. L#2, on the other hand, directly provides the answer. After learning from L#2's response, X#3 adopts the same answering style and provides the correct answer. This demonstrates that UML, even with limited unlabeled data, can enhance LLM performance through cross-model response learning. This finding further supports the second mechanism of UML: increased response diversity enables effective model

refinement. It is also noteworthy that although L#2's answer for the sample from the train set is incorrect, X#3 correctly answers the question from the test set. This finding strongly supports the first mechanism of UML: samples with incorrect labels have the potential to improve model performance.

IV. Conclusion and limitations

In this paper, we introduce UML, a novel framework that enables mutual learning among small and comparable models. Empirically, UML outperforms 8 baselines on 4 diverse tasks. To understand the success, we conduct ablation studies, which confirm that the underlying mechanisms of UML are twofold: LLMs are capable of unsupervised learning, and they can benefit from response diversity. Further analysis reveals that model diversity is the key driver, underscoring the potential of mutual learning among small and comparable models as a powerful alternative to reliance on large models. Finally, we explore the effects of key design choices, including selection strategies, learning orders, iteration round and train sizes. Meanwhile, we acknowledge that UML is limited to text-based data and cannot integrate multi-modal information. In the future, we will extend UML to incorporate vision modality.

V. Acknowledgments

The paper was completely written by human. The text was subsequently polished using DeepSeek, ChatGPT, and Qwen. This paper was supported by the Shanghai Natural Science Foundation Project under Grant 25ZR1402116.

References

[1] W. Zhang, W. Lu, J. Wang, Y. Wang, L. Chen, H. Jiang, J. Liu, and T. Ruan, "Unexpected phenomenon: LLMs' spurious associations in information extraction," in *Findings Assoc. Comput. Linguistics: ACL 2024*, Aug. 2024, pp. 9176–9190.

[2] H. Yang, H. Chen, H. Guo, Y. Chen, C.-S. Lin, S. Hu, J. Hu, X. Wu, and X. Wang, "Llm-medqa: Enhancing medical question answering through case studies in large language models," in *2025 Int. Joint Conf. Neural Networks (IJCNN)*, 2025, pp. 1–8.

[3] X. Zhu, J. Li, R. Yin, C. Ma, and W. Wang, "Improving mathematical reasoning capabilities of small language models via feedback-driven distillation," in *2025 Int. Joint Conf. Neural Networks (IJCNN)*, 2025, pp. 1–10.

[4] Y. Annepaka and P. Pakray, "Large language models: a survey of their development, capabilities, and applications," *Knowl. and Inf. Systems*, vol. 67, no. 3, pp. 2967–3022, Mar 2025.

[5] M. Tao, D. Zhao, and Y. Feng, "Chain-of-discussion: A multi-model framework for complex evidence-based question answering," in *Proc. 31st Int. Conf. Comput. Linguistics*, Jan. 2025, pp. 11 070–11 085.

[6] S. Feng, W. Shi, Y. Wang, W. Ding, V. Balachandran, and Y. Tsvetkov, "Don't hallucinate, abstain: Identifying LLM knowledge gaps via multi-LLM collaboration," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, Aug. 2024, pp. 14 664–14 690.

[7] C.-M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang *et al.*, "Chateval: Towards better LLM-based evaluators through multi-agent debate," in *12th Int. Conf. Learn. Representations*, 2024.

[8] J. Lu, W. Zhong, Y. Wang, Z. Guo, Q. Zhu, W. Huang *et al.*, "Yoda: Teacher-student progressive learning for language models," *CoRR*, vol. abs/2401.15670, 2024.

[9] M. Li, L. Chen, J. Chen, S. He, J. Gu, and T. Zhou, "Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning," in *Findings Assoc. Comput. Linguistics ACL 2024*, Aug. 2024, pp. 16 189–16 211.

[10] T. Kuzman and N. Ljubešić, "Llm teacher-student framework for text classification with no manually annotated data: A case study in iptc news topic classification," *IEEE Access*, vol. 13, pp. 35 621–35 633, 2025.

[11] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, and A. Miller, "Language models as knowledge bases?" in *Proc. 2019 Conf. Empirical Methods Natural Language Process. 9th Int. Joint Conf. Natural Language Process. (EMNLP-IJCNLP)*, Nov. 2019, pp. 2463–2473.

[12] S. Lin, J. Hilton, and O. Evans, "TruthfulQA: Measuring how models mimic human falsehoods," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, May 2022, pp. 3214–3252.

[13] L. Zheng, N. Guha, B. R. Anderson, P. Henderson, and D. E. Ho, "When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings," in *Proc. 18th Int. Conf. Artif. Intell. Law*, 2021, pp. 159–168.

[14] X. Li and D. Roth, "Learning question classifiers," in *COLING 2002: 19th Int. Conf. Comput. Linguistics*, 2002.

[15] X. Hu, D. Li, B. Hu, Z. Zheng, Z. Liu, and M. Zhang, "Separate the wheat from the chaff: model deficiency unlearning via parameter-efficient module operation," in *Proc. 38th AAAI Conf. Artif. Intell. and 36th Conf. Innov. Appl. Artif. Intell. and 14th Symp. Educ. Adv. Artif. Intell.*, ser. AAAI'24/IAAI'24/EAAI'24, 2024.

[16] E. Hovy, L. Gerber, U. Hermjakob, C.-Y. Lin, and D. Ravichandran, "Toward semantics-based answer pinpointing," in *Proc. 1st Int. Conf. Human Language Technology Research*, 2001.

[17] J. Mohta, K. Ak, Y. Xu, and M. Shen, "Are large language models good annotators?" in *Proc. "I Can't Believe It's Not Better: Failure Modes in the Age of Foundation Models" NeurIPS 2023 Workshops*, vol. 239, 16 Dec 2023, pp. 38–48.

[18] W. Chen, D. Song, and B. Li, "GRATH: Gradual self-truthifying for large language models," in *Proc. 41st Int. Conf. Mach. Learn.*, vol. 235, 21–27 Jul 2024, pp. 7260–7279.

[19] J. Yeom, H. Lee, H. Byun, Y. Kim, J. Byun, Y. Choi *et al.*, "Tc-llama 2: fine-tuning llm for technology and commercialization applications," *J. Big Data*, vol. 11, no. 1, p. 100, Aug 2024.

[20] M. Chartier, N. Dakkoune, G. Bourgeois, and S. Jean, "Hibenchllm: Historical inquiry benchmarking for large language models," *Data & Knowl. Engineering*, vol. 156, p. 102383, 2025.

[21] J. Xiao, Y. Chen, Y. Ou, H. Yu, K. Shu, and Y. Xiao, "Baichuan2-sum: Instruction finetune baichuan2-7b model for dialogue summarization," in *2024 Int. Joint Conf. Neural Networks (IJCNN)*, 2024, pp. 1–8.

[22] Z. Mai, J. Zhang, Z. Xu, and Z. Xiao, "Financial sentiment analysis meets llama 3: A comprehensive analysis," in *Proc. 2024 7th Int. Conf. Mach. Learn. Mach. Intell. (MLMI)*, 2024, p. 171–175.

[23] Z. Shen, H. Lang, B. Wang, Y. Kim, and D. Sontag, "Learning to decode collaboratively with multiple language models," in *Proc. 62nd Annu. Meeting Assoc. Comput. Linguistics (Volume 1: Long Papers)*, Aug. 2024, pp. 12 974–12 990.

[24] X. Ning, Z. Wang, S. Li, Z. Lin, P. Yao, T. Fu, M. B. Blaschko, G. Dai, H. Yang, and Y. Wang, "Can llms learn by teaching for better reasoning? a preliminary study," in *Advances Neural Inf. Process. Systems*, vol. 37, 2024, pp. 71 188–71 239.

[25] Y. Liu, H. He, T. Han, X. Zhang, M. Liu, J. Tian, Y. Zhang, J. Wang, X. Gao, T. Zhong, Y. Pan, S. Xu, Z. Wu, Z. Liu, X. Zhang, S. Zhang, X. Hu, T. Zhang, N. Qiang, T. Liu, and B. Ge, "Understanding llms: A comprehensive overview from training to inference," *Neurocomputing*, vol. 620, p. 129190, 2025.

[26] Q. Zhang, M. Chen, A. Bukharin, P. He, Y. Cheng, W. Chen *et al.*, "Adaptive budget allocation for parameter-efficient fine-tuning," in *11th Int. Conf. Learn. Representations*, 2023.

[27] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Int. Conf. Learn. Representations*, 2022.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Systems*, 2017, p. 6000–6010.

[29] S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi *et al.*, "Rethinking the role of demonstrations: What makes in-context learning work?" in *Proc. 2022 Conf. Empirical Methods Natural Language Process.*, Dec. 2022, pp. 11 048–11 064.

[30] T. Schick and H. Schütze, "Exploiting cloze-questions for few-shot text classification and natural language inference," in *Proc. 16th Conf. European Chapter Assoc. Comput. Linguistics: Main Volume*, Apr. 2021, pp. 255–269.

[31] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal, "Detecting hallucinations in large language models using semantic entropy," *Nature*, vol. 630, no. 8017, pp. 625–630, Jun 2024.