SRJudge: Empowering Large Language Models with Selective Reasoning for Fine-Grained Knowledge Concept Tagging

Zhiwei Yang[1] , **Jiahua Yang**[1] , **Huiru Lin**[2,3] and **Xing Chen**[4,*] and **Quanlong Guan**[1,*]
[1]Guangdong Institute of Smart Education, Jinan University, Guangzhou, China
[2]School of Physical Education, Jinan University, Guangzhou, China
[3]Guangdong Provincial Key Laboratory of Speed Capability Research, Guangzhou, China
[4]Sapient Intelligence Pte Ltd, Singapore
{yangzw, linhuiru, gql}@jnu.edu.cn, yangjiahua@stu2024 jnu.edu.cn, raincchio@gmail.com

Abstract

Knowledge concept tagging aims to assign specific concept or topic labels to educational content, which is essential for both educators and learners in traditional and online teaching practices. Recent work has explored large language models (LLMs) for this task, achieving promising performance. However, LLMs still struggle to select the correct concept from a large-scale candidate set due to the high dimensionality of the decision space. In this paper, we propose a novel three-stage **S**elect-**R**eason-**Judge** (SRJudge) framework, which empowers LLMs with selective reasoning capability for fine-grained knowledge concept tagging. Specifically, the Selector in Stage 1 first narrows the candidate concepts to a top-K shortlist by fine-tuning a small language model (SLM), e.g., BERT, since the top-K predictions hit the correct concept in most cases, thereby reducing the decision space of correct candidates. Next, the Stage 2 Reasoner employs a lightweight LLM for refined reasoning over the shortlisted candidates. It further integrates an improved reinforcement learning strategy with a dynamic task-specific reward function and a pruning mechanism to better align with human reasoning preferences. Finally, a larger LLM acts as a judger that evaluates the overall rationality of the reasoning process and its explanations to determine the final output. In addition, we construct two high-quality datasets for further validation, i.e., the biology dataset **S_Bio** and the physics dataset **S_Phy**. Experimental results demonstrate that our method consistently outperforms state-of-the-art baselines across benchmark datasets, verifying its effectiveness and superiority. Resources are available at: https://github.com/Nicozwy/SRJudge.

1 Introduction

Knowledge concepts form the foundation for key educational tasks on online platforms, such as MOOCs and intelligent tutoring systems. These tasks include knowledge tracing [Cui *et al.*, 2024], cognitive diagnosis [Wang *et al.*, 2024], and intelligent recommendation [Ozyurt *et al.*, 2025]. To accurately index educational materials, it requires annotators to identify appropriate concepts based on the semantic and logical information within the question and its solution, as shown in Figure 1. Most existing knowledge concept tagging methods perform well yet struggle to select the correct concept among numerous similar candidates, making this a key performance bottleneck. Given the strong reasoning capabilities of large language models (LLMs), they could serve as enhanced tools for contextual understanding and disambiguation in knowledge concept tagging. However, LLMs often face challenges in selecting the correct concept from an extensive candidate set due to the high dimensionality of the decision space. Thus, the research on equipping LLMs with effective reasoning capabilities for navigating large-scale candidate concept sets is both challenging and essential.

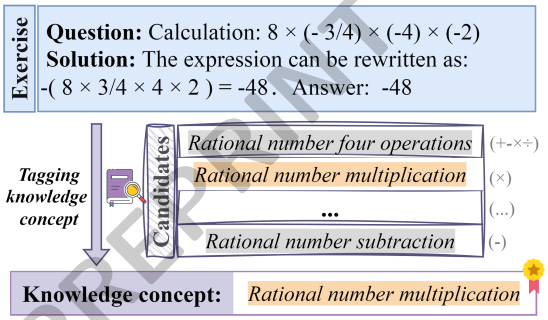

Figure 1: Illustration of the knowledge concept tagging task.

Existing tagging methods achieve decent performance utilizing support vector machines (SVM) [Saha and Schneider, 2013], recurrent neural networks (RNNs) [Sun *et al.*, 2018], graph convolutional networks (GCNs) [Zhang *et al.*, 2021], and RoBERTa [Ding *et al.*, 2025a]. However, they are limited in capturing effective features and reasoning about similar knowledge. With the emergence of large language models (LLMs) [Achiam *et al.*, 2023; Bai *et al.*, 2023; Touvron *et al.*, 2023], they have demonstrated impressive performance across various domains, such as mathematical reasoning [Xia *et al.*, 2025], code generation [Gu, 2023], medicine

*Corresponding authors: Xing Chen, Quanlong Guan

Model	S_Math		S_Bio		S_Phy	
	ACC	ACC@5	ACC	ACC@5	ACC	ACC@5
Chinese-BERT	0.7200	0.9107	0.7216	**0.9377**	0.6344	**0.9154**
Chinese-RoBERTa	0.7191	0.9148	0.7176	0.9247	0.6352	0.9100
Conan-embedding	0.7240	**0.9156**	0.7226	0.9296	0.6367	0.9051

Table 1: Performance of BERT models with *supervised fine-tuning*, where ACC@5 denotes the accuracy of including the correct answer among the top 5 predictions.

[Thirunavukarasu *et al.*, 2023], etc. Recently, [Ozyurt *et al.*, 2024] leveraged LLMs' prior knowledge to generate step-by-step solutions for problems, then tagged each step with corresponding knowledge concepts. [Moore *et al.*, 2024] designed two different prompting strategies to guide GPT-4 in generating knowledge concepts for multiple-choice questions, demonstrating LLMs can generate accurate knowledge concepts. To determine whether an existing knowledge concept aligns with a given question, [Li *et al.*, 2025] reformulated the tagging process into sub-problems and utilized collaborative LLM-based agents to improve tagging consistency and robustness. In addition, [Yang *et al.*, 2025] employed cascaded soft voting to integrate predictions from BERT-based models and LLMs, highlighting their complementary strengths. However, the large candidate set increases the dimensionality of the decision space, making it harder for the model to identify the relevant features that distinguish the correct concept.

As our preliminary studies shown in Table 1, traditional SLMs can effectively capture the correct options within the top K shortlist, but their accuracy remains limited when relying solely on the top-1 prediction. This arises from the semantic similarity among concepts within the top K candidates. In contrast, while LLMs exhibit stronger generation and reasoning capabilities, a gap remains between model and human reasoning, particularly in selecting from large-scale candidates. Thus, these inherent limitations hinder existing methods from improving tagging performance.

To this end, we propose a novel three-stage Select-Reason-Judge (SRJudge) framework that enables LLMs to effectively reason over large concept sets to achieve accurate tagging. Specifically, we first fine-tune a BERT-based Selector to select K candidate concepts (Stage 1). Then, we apply pruning-based optimization on the generalized reweighted policy optimization (GRPO) method to train a lightweight LLM Reasoner on the K coarse-grained shortlist, and then design a task-specific reward function that introduces an additional dynamic position reward for the shortlist from Stage 1, reinforcing the Reasoner to recommend the most suitable concept along with its reasons (Stage 2). Finally, we incorporate an LLM Judger to evaluate the rationality of the recommendations and reasons from a global perspective and provide the final verdict (Stage 3). Moreover, since no other validation datasets are available, we built two new datasets for method evaluation. Experimental results on datasets from three disciplines demonstrate that SRJudge consistently outperforms existing methods, validating the effectiveness of selective reasoning with reinforcement learning in enabling LLMs to tag the correct concept from a large candidate set.

Our contributions are summarized as follows:

- We present a novel three-stage framework, Select-Reason-Judge (SRJudge), that equips LLMs with selective reasoning capabilities through reinforcement learning for knowledge concept tagging amidst numerous candidates.
- We construct two new datasets, i.e., S_Bio and S_Phy, which will be made available to fill the gap in cross-validation of knowledge concept tagging methods across different disciplines.
- Experimental results show that SRJudge significantly outperforms leading baselines, highlighting its ability to enhance LLMs with selective reasoning to distinguish fine-grained knowledge concepts across different disciplines.

2 Related Work

2.1 Traditional Knowledge Concept Tagging

Traditional tagging methods primarily rely on traditional machine learning, lightweight deep learning, and small-scale pretrained models, involving feature engineering, contextual modeling, and graph structures [Li *et al.*, 2022]. For example, [Saha and Schneider, 2013] constructed a discriminative model based on SVM to perform tagging on questions from question-answering (QA) websites. Subsequently, deep learning methods were introduced into this domain. To enhance tagging performance, [Sun *et al.*, 2018] employed RNNs to capture contextual dependencies. To further address complex semantic and structural information, [Zhang *et al.*, 2021] proposed a graph-guided topic tagging model that integrates directed acyclic graphs and multi-dimensional attention to precisely tag topics in the QA dataset. Building upon this line of research, [Yang *et al.*, 2022] modeled sentences and concept graphs through graph-to-graph matching, mining logical relations and concept correspondences among terms using relational graph convolutional networks. [Liu *et al.*, 2024] improved graph contrastive learning to obtain clearer and more discriminative class boundaries. [Huang *et al.*, 2023] jointly modeled questions and answers using BERT and fused features to enhance the tagging performance.

Although traditional methods have achieved successes in knowledge concept tagging, they still suffer from limited capabilities in capturing effective features and reasoning about similar knowledge, hindering their effectiveness in fine-grained knowledge concept tagging scenarios.

2.2 LLM-based Tagging

Recently, LLM-based tagging methods have explored the robust generative and reasoning capabilities of LLMs to enhance tagging accuracy and interpretability through prompt engineering [Moore *et al.*, 2024], CoT reasoning [Ozyurt *et al.*, 2024], and multi-agent collaboration [Li *et al.*, 2025]. For example, [Ozyurt *et al.*, 2024] improved the performance of knowledge tracing models by utilizing LLMs first to solve exercises via CoT, followed by automatic tagging of knowledge concepts with these exercises, thereby strengthening the semantic linkage between exercises and knowledge concepts. Building on this, [Moore *et al.*, 2024] utilized two different prompting strategies to guide GPT-4 to generate a knowledge

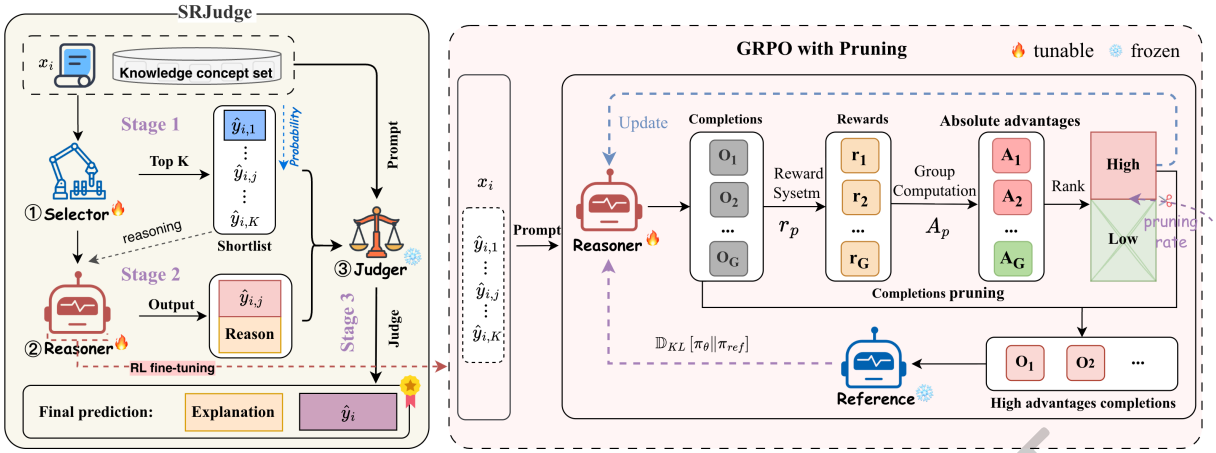

Figure 2: The proposed SRJudge framework for fine-grained knowledge concept tagging. RL denotes reinforcement learning, "Reference" denotes a frozen counterpart of the policy model (i.e., LLM Reasoner), O_p is a binary pair defined as $O_p = (\hat{y}_{i,j}, \psi_{i,j}), 1 \leq p \leq G$.

concept for multiple-choice questions and compares them with the original knowledge concepts. Results revealed a human preference for LLM-generated concepts over original annotations, highlighting the superior generative quality of LLMs in this context. To address the complexity and improve the reliability of exercise tagging, [Li *et al.*, 2025] reformulated the exercise tagging judgment process as a collaboration among multiple LLM-based agents tackling independent subproblems. Prior work indicates that collaborative model ensemble strategies improve text classification performance. Specifically, [Zhang *et al.*, 2025] reported substantial gains from iteratively ensembling seven strong LLMs. Similarly, [Yang *et al.*, 2025] reported that integrating fine-tuned BERT-series models with LLMs through local and global ensemble voting significantly improves labeling performance. Although fine-tuned LLMs achieve decent performance, they may not cover all potential user needs with static datasets, suffering from high computational costs [Ghosh *et al.*, 2024]. Additionally, prompt-based reasoning struggles to align with human reasoning preferences.

3 Proposed Method

3.1 Problem Statement

Given an exercise $x_i \in \{x_1, x_2, \ldots, x_m\}$ associated with a knowledge concept label $y_i \in \Phi = \{y_i\}_{i=1}^{|\Phi|}$, the task of knowledge concept tagging aims to predict the label $\hat{y}_i$ for x_i, i.e., $f(x_i) \longrightarrow \hat{y}_i$, where $|\Phi|$ represents the category number of candidate knowledge concepts.

Overview: As illustrated in Figure 2, our proposed SRJudge comprises three key components: Selector, Reasoner, and Judger. For the exercise x_i, the Selector first picks $K(\ll |\Phi|)$ candidates, $\{\hat{y}_{i,1}, \ldots, \hat{y}_{i,j}, \ldots, \hat{y}_{i,K}\}$, reducing the decision space. The Reasoner then performs reasoning over the shortlist, producing both the recommended concept $\hat{y}_{i,j}$ and corresponding reasons. Finally, the Judger assesses the overall rationality of the reasoning process and its explanations to determine the final output $\hat{y}_i$.

3.2 Selector for Scope Reduction

To reduce the candidate concept space, we use a lightweight pre-trained model (e.g., BERT) to select the top K candidates from a large pool in Stage 1, thereby improving exploration efficiency while maximizing retention of correct concepts in the shortlist. In this work, we employ unsupervised pre-training and supervised fine-tuning to achieve a more effective Selector. Specifically, masked language modeling is employed in the continued pre-training phase to enhance the model's semantic understanding. We fine-tune the Selector on a labeled dataset to improve its accuracy in determining the relevance of knowledge concepts to a given query. Thus, these K candidates are ranked and selected based on predicted probabilities using softmax normalization as follows:

$$\mathcal{C}_{\text{top } K} = \{\hat{y}_{i,j} \in \Phi \mid \text{rank}_{\downarrow}(p_{i,j}) \leq K\} \tag{1}$$

where $\text{rank}_{\downarrow}()$ indicates ranking candidates in descending order, $p_{i,j}$ is the probability of candidate $\hat{y}_{i,j}$. This selected candidate set $\mathcal{C}_{\text{top } K}$ is then fed into Stage 2 to ensure that higher-quality candidates are concentrated within a smaller scope, facilitating subsequent reinforcement learning (RL).

3.3 Reasoner for Concept Recommendation

In Stage 2, we employ an RL algorithm — Group Relative Policy Optimization(GRPO) [Shao *et al.*, 2024] to enhance the reasoning accuracy of LLMs on the K candidates for fine-grained knowledge concept tagging. That is, the candidate concept set $\{\hat{y}_{i,1}, \ldots, \hat{y}_{i,j}, \ldots, \hat{y}_{i,K}\}$ provided by the Selector is treated as the decision space. For training efficiency, we utilize a lightweight LLM (e.g., Qwen2.5-1.5B-Instruct) as the policy model and apply GRPO to guide accurate concept tagging with corresponding justifications, employing a newly designed task-specific reward function. The goal is to select the most suitable concept label $\hat{y}_{i,j}$ for each exercise and provide the corresponding reasons. Specifically, for each exercise x ($x \sim P(Q)$, omit the subscript i for convenience), GRPO samples G output completions $\{o_1, o_2, \ldots, o_p, \ldots, o_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$, where o_p is defined as $o_p = (\hat{y}_{i,j}, \psi_{i,j})$, where $\hat{y}_{i,j}$ and $\psi_{i,j}$ denote the

predicted knowledge concept and reasons in the p-th output, respectively. Then the policy model is optimized by maximizing the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim P(Q),\{o_p\}_{p=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left\{ \frac{1}{G}\sum_{p=1}^{G}\frac{1}{|o_p|}\sum_{t=1}^{|o_p|} \left\{ \min\left[\frac{\pi_\theta(o_{p,t}|x,o_{p,<t})}{\pi_{\theta_{\text{old}}}(o_{p,t}|x,o_{p,<t})}\hat{A}_{p,t}, \right.\right.\right. \\ \left.\left.\left. \text{clip}(\frac{\pi_\theta(o_{p,t}|x,o_{p,<t})}{\pi_{\theta_{\text{old}}}(o_{p,t}|x,o_{p,<t})}, 1-\varepsilon, 1+\varepsilon)\hat{A}_{p,t} \right] - \beta\mathbb{D}_{KL}\left[\pi_\theta\|\pi_{ref}\right] \right\} \right\} \tag{2}$$

where θ denotes the parameters of the current policy model π_θ, π_{ref} is the reference policy model, ε is a clipping-related hyper-parameter introduced in GRPO for stabilizing training, and β is the coefficient of the KL penalty. $\hat{A}_{p,t}$ is the advantage value computed based solely on the relative rewards within each group of outputs. GRPO regularizes the policy update by directly adding the Kullback–Leibler (KL) [Kullback and Leibler, 1951] divergence between the training policy and the reference policy to the loss function. A_p is the advantage computed using a group of rewards $\{r_1, r_2, \ldots, r_p, \ldots, r_G\}$ corresponding to the completions within each group:

$$A_p = \frac{r_p - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})} \tag{3}$$

Task-specific Reward. To leverage GRPO for enhancing LLM reasoning on fine-grained concepts, the reward function is specifically designed to optimize task performance by incorporating three types of rewards: output format, answer accuracy, and correct answer position, as follows:

$$r_p = R_{\text{format}}(o_p) + R_{\text{acc}}(o_p) + R_{\text{position}}(o_p) \tag{4}$$

$R_{\text{format}}(o_p)$ denotes the format reward to enforce a standardized output format, requiring the final answer to be enclosed within `\boxed{}` (+1) for answer extraction. The model's reasoning process must be clearly marked using the tags `<reason>` and `</reason>` (+0.5) to ensure a well-structured explanation. $R_{\text{acc}}(o_p)$ represents the accuracy reward derived from the correctness of the model's answer, providing positive feedback when the answer is correct (+3) and no reward otherwise. Additionally, $R_{\text{position}}(o_p)$ denotes the position reward, which is dynamically **conditioned on the model predicting correctly and and the correct answer being ranked at position** $r \in [1, K-1]$ **in the candidate shortlist**. This is formulated as follows:

$$R_{\text{position}}(o_p) = \alpha \cdot \frac{\frac{1}{p_r}}{\sum_{j=0}^{K-1}\frac{1}{p_j}} \cdot \frac{\log_2(r+1)}{\log_2 K}, \\ \alpha = \frac{10 \times \text{S}_\text{c}}{\text{S}_\text{T} \times (1 - \text{pruning rate})} + 1 \tag{5}$$

where r is the position of the correct answer in the candidate shortlist (starting from 0), K is the length of the shortlist, p_r is the probability associated with rank r, and α is a dynamic scaling factor determined by the current training step and model pruning rate. Specifically, S_c and S_T represent the current training step and the total number of training steps, respectively. The proposed position-based reward is designed to ensure bounded and stable policy gradients. The normalization of $R_{\text{position}}(o_p)$ within a manageable range prevents excessively large gradients during early training, which effectively improves convergence stability. A small α at the beginning helps reduce gradient variance and stabilize learning, while a larger α in the later phase enhances reward diversity and prevents premature convergence. Since candidates are ranked by confidence, their positions carry meaningful prior information. To encourage moderate exploration, the additional position reward is granted only when the correct answer appears between the 2nd and K-th positions ($r \in [1, K-1]$). When the correct answer is at the first position ($r = 0$), the logarithmic term $\log_2(r+1)$ evaluates to zero, and thus $R_{\text{position}} = 0$. This design maintains training stability in the early stages while gradually encouraging exploration of lower-ranked candidates as training progresses. Thus, it balances exploration and exploitation, stabilizing policy gradient updates throughout the learning process.

Pruning Mechanism. When applying GRPO to knowledge concept tagging, the agent can be distracted by low-quality completions with positive relative advantages. During RL training, such low-quality outputs typically manifest as "format errors with correct answers" or "correct formats with incorrect answers" in the early stages, and predominantly as "correct formats with incorrect answers" in later stages. These completions introduce noise that hinders effective agent–environment interaction, and ultimately degrades the stability and efficiency of policy optimization. Therefore, we introduce a pruning mechanism in reinforcement learning, which computes the absolute advantage of each completion, ranks them, and dynamically filters out those with low absolute advantage. To facilitate model learning, we further impose the confidence constraint on the GRPO optimization objective in Equation (2), as follows:

$$|\hat{A}_{p,t}| \geq \gamma \tag{6}$$

where γ is selected from the set of absolute advantages $\{|\hat{A}_{p,t}|\}$ according to the predefined pruning rate. Specifically, γ represents the minimum absolute advantage that a completion among the G generated outputs must satisfy to be retained for gradient updates. This ensures that only completions with strong training signals are preserved for policy updates, thereby improving training efficiency and mitigating interference from ineffective or low-quality samples.

3.4 Judger for Overall Assessment

In Stage 3, we utilize a frozen LLM (e.g., Qwen3-32B) as the Judger to comprehensively evaluate and select the correct knowledge concept and generate explanations after reinforced reasoning, which further enhances the accuracy and robustness of the final decision from a global perspective. Specifically, the Judger takes as input the top 1 prediction from the Selector, the selected K candidate concepts, the recommended concepts, and the corresponding reasons gener-

ated by the Reasoner. Thus, the Judger makes final decisions from global perspectives utilizing the following prompts:

System-prompt: You are an assistant for tagging knowledge concepts in mathematics exercises and do not need to solve the problem. In the following dialogue, I would like you to reason step by step. Please refer to both the predictions and reasoning from the two models, as well as the list of knowledge concepts, to help me select the knowledge concept that best fits the question. Please enclose the selected knowledge concept within boxed $\{\hat{y}_i\}$.

User-prompt:" The mathematics question and its solution are: [from the dataset]

The list of knowledge concepts is: [from Stage 1]
Model 1 predicted_label: [from Stage 1]
Model 2 predicted_label: [from Stage 2]
The reason for Model 2's choice is: [from Stage 2] ***"***

4 Experiments

4.1 Datasets and Baseline Methods

To verify the effectiveness of SRJudge, we conduct experiments on three educational datasets: **S_Math** [Yang *et al.*, 2025], **S_Bio**, and **S_Phy**, For cross-validation, we constructed the new biological dataset, S_Bio, and the new physics dataset, S_Phy. Both datasets were developed in a similar manner, incorporating questions, solutions, difficulty scores, and knowledge concepts. For unsupervised training, there are 23,596, 15,669, and 32,138 unlabeled instances in U_Math, U_Bio, and U_Phy, respectively (See our repository for more details). We perform an 8:1:1 stratified split into training, validation, and test sets for each dataset, ensuring that the distribution of concept categories is balanced across each subset. Table 2 provides the dataset statistics.

Baseline Methods: We compare our proposed framework with the state-of-the-art baselines as follows:

- SLM-based methods: We respectively compare the use of BERT [Devlin *et al.*, 2019], RoBERTa [Liu *et al.*, 2019], and Conan-Embedding [Li *et al.*, 2024] as backbone encoders, each of which is followed by a classification head and trained via supervised fine-tuning (SFT).
- LLM-based methods: We employ mainstream LLMs, including Qwen [Bai *et al.*, 2023], LLaMA [Touvron *et al.*, 2023], DeepSeek (DS) [Guo *et al.*, 2025], and ChatGPT [Achiam *et al.*, 2023]. For LLMs with fewer than 8B parameters, we explore SFT and $GRPO$ strategies, respectively. Due to computational cost, we adopt SFT for open-source models and zero-shot CoT prompting (COT) for closed-source models for LLMs with over 8B parameters.
- Additional Baselines: Among existing studies on exercise knowledge concept tagging, PQSCT [Huang *et al.*, 2023], LGCEL [Yang *et al.*, 2025], and LHABS [Ding *et al.*, 2025b] are specifically designed for this task. PQSCT jointly models questions and answers using BERT with feature fusion techniques, LGCEL leverages multiple heterogeneous expert models to enhance tagging accuracy through model complementarity, while LHABS improves RoBERTa with extra attention and label smoothing. We further compare two latest classification baselines: GIFT [Liu *et al.*, 2024] employs a graph neural network combined with contrastive learning, and RGPT [Zhang *et al.*, 2025] integrates seven LLaMA models through a cascaded guidance mechanism for text classification.

Dataset	Sum	Train	Valid	Test	Category	Level
S_Math	12,205	9,763	1,221	1,221	155	G7–G9
S_Bio	9,941	7,952	994	995	72	G10–G12
S_Phy	18,440	14,752	1,844	1,844	135	G10–G12

Table 2: Dataset statistics.

Experimental Settings: In supervised fine-tuning, we set the batch size, number of epochs, and learning rate to 16, 15, and 1e-4, respectively. Evaluation is performed every 100 steps, with early stopping applied (patience = 10). For GRPO, the batch size, number of epochs, number of generations, and pruning rate are set to 12, 3, 12, and 0.5, respectively. Additionally, the clipping parameter ε, the KL penalty coefficient β, and the learning rate are set to 0.2, 0.04, and 1×10^{-6}, respectively. The hyperparameter K is set to 5. Our proposed method is implemented and evaluated on a cluster of four NVIDIA L20 servers, using accuracy (**ACC**), precision (**P**), recall (**R**), and macro F1 score (**F1**) as evaluation metrics.

4.2 Overall Results and Analysis

Table 3 shows the experimental results of SRJudge against various baselines on S_Math, S_Bio, and S_Phy. Overall, SRJudge outperforms LGCEL, achieving an F1 score of 0.7602 on S_Math, 0.6987 on S_Bio, and 0.6643 on S_Phy, consistently yielding the best results. This demonstrates the effectiveness of the proposed three-stage framework, SRJudge, which reduces the decision space through supervised fine-tuning, reinforces LLM reasoning via tailored GRPO, and provides an overall assessment from a global perspective.

Specifically, the latest baselines, including PQSCT, GIFT, LHABS, and RGPT, demonstrate relatively unsatisfactory performance compared to LGCEL, which remains the leading baseline, achieving F1 scores of 0.7311, 0.6840, and 0.6476 on the S_Math, S_Bio, and S_Phy datasets, respectively. This is because PQSCT fuses separate features of questions and answers without exploring more effective information, while GIFT may introduce noise by depending on external knowledge graphs for augmented view generation. Furthermore, LHABS shows reduced performance, likely due to label smoothing, which diminishes subtle distinctions among knowledge concepts, thereby complicating the model's ability to differentiate between them. Although RGPT shares conceptual similarities with LGCEL, it suffers from limited model diversity. In contrast, LGCEL leverages heterogeneous model predictions that complement one another, thereby enhancing overall performance.

Moreover, we conduct a comprehensive comparison of more SLMs and LLMs for in-depth analysis. SLM-based tagging methods using the SFT strategy achieve lower performance compared to LLM-based methods, but due to their smaller parameter size and lower training cost, they remain highly practical. For LLMs employing the SFT strategy, Qwen2.5-72B-Instruct outperforms Qwen2.5-7B-

Model	Strategy	S_Math				S_Bio				S_Phy			
		ACC	P	R	F1	ACC	P	R	F1	ACC	P	R	F1
Chinese-RoBERTa	SFT	0.7191	0.7213	0.7062	0.6992	0.7176	0.6889	0.6624	0.6613	0.6352	0.6378	0.6232	0.6227
Chinese-BERT	SFT	0.7200	0.7213	0.6938	0.6911	0.7216	0.6900	0.6743	0.6682	0.6344	0.6359	0.6217	0.6203
Conan-embedding	SFT	0.7240	0.7164	0.7118	0.7017	0.7226	0.7003	0.6725	0.6786	0.6367	0.6267	0.6177	0.6165
Qwen2.5-7B-Instruct	SFT	0.7428	0.7273	0.7165	0.7041	0.7286	0.6989	0.6808	0.6790	0.6524	0.6555	0.6340	0.6336
	$GRPO$	0.7117	0.6989	0.6833	0.6600	0.6613	0.5874	0.5923	0.5674	0.6410	0.5721	0.5598	0.5388
LLaMA-3.1-8B-Instruct	SFT	0.7371	0.7223	0.7185	0.7079	0.7266	0.6967	0.6775	0.6794	0.6534	0.6537	0.6276	0.6312
	$GRPO$	0.6617	0.6551	0.6530	0.6278	0.6119	0.5601	0.5719	0.5398	0.6133	0.5523	0.5331	0.5250
DS-R1-Qwen-7B	SFT	0.7428	0.7415	0.7192	0.7141	0.7226	0.7092	0.6748	0.6739	0.6504	0.6470	0.6272	0.6266
	$GRPO$	0.6642	0.6108	0.6132	0.5917	0.6442	0.5482	0.5772	0.5410	0.6318	0.5706	0.5584	0.5345
DS-R1-Qwen-32B	SFT	0.7387	0.7218	0.7193	0.7175	0.7256	0.7024	0.6797	0.6752	0.6947	0.6577	0.6419	0.6390
DS-R1-LLaMA-72B	SFT	0.7477	0.7305	0.7248	0.7264	0.7367	0.7081	0.6796	0.6821	0.6795	0.6742	0.6558	0.6586
Qwen3-14B	SFT	0.7461	0.7249	0.7200	0.7156	0.7387	0.7018	0.6847	0.6818	0.6676	0.6554	0.6418	0.6366
Qwen3-32B	SFT	0.7494	0.7456	0.7175	0.7187	0.7387	0.7069	0.6828	0.6841	0.6714	0.6708	0.6449	0.6447
Qwen2.5-72B-Instruct	SFT	0.7518	0.7517	0.7188	0.7204	0.7396	0.7127	0.7013	0.6855	0.6768	0.6709	0.6585	0.6526
DS-R1-0528	COT	0.6355	0.6458	0.6239	0.5954	0.6391	0.5876	0.5981	0.5747	0.4506	0.4596	0.4206	0.3923
GPT-4.1	COT	0.5659	0.5942	0.5350	0.5047	0.5728	0.5311	0.5288	0.5068	0.4192	0.4115	0.4040	0.3664
PQSCT	–	0.7174	0.7182	0.6937	0.6896	0.7296	0.6935	0.6716	0.6667	0.6676	0.6525	0.6245	0.6244
GIFT	–	0.7027	0.6986	0.6824	0.6749	0.7256	0.6939	0.6821	0.6758	0.6649	0.6641	0.6285	0.6210
LHABS	–	0.7158	0.7091	0.6969	0.6864	0.7327	0.6939	0.6852	0.6763	0.6708	0.6543	0.6251	0.6229
RGPT	–	0.7412	0.7215	0.7211	0.7086	0.7236	0.6892	0.6833	0.6712	0.6703	0.6509	0.6353	0.6287
LGCEL	–	0.7660	0.7602	0.7379	0.7311	0.7417	0.7042	0.6831	0.6840	0.6725	0.6700	0.6489	0.6476
SRJudge (Ours)	–	**0.7748**	**0.7772**	**0.7721**	**0.7602**	**0.7437**	**0.7285**	**0.7029**	**0.6987**	**0.6920**	**0.6791**	**0.6702**	**0.6643**

Table 3: Performance comparison on the S_Math, S_Bio, and S_Phy datasets. The bold numbers denote the best results, and the underlined numbers denote the best performance of baselines using different strategies (statistically significant at $p < 0.05$).

Instruct, suggesting that increasing model size can enhance their logical reasoning and tagging performance to some extent. In the COT strategy setting, even state-of-the-art models such as DeepSeek-R1-0528 and GPT-4.1 struggle to achieve satisfactory results due to the large decision space and the lack of supervised fine-tuning. While LLMs can benefit from $GRPO$ in knowledge concept tagging, their performance remains significantly inferior to that achieved with the SFT strategy, obtaining F1 scores of only 0.6600, 0.5674, and 0.5388 on the S_Math, S_Bio, and S_Phy datasets, respectively. This is also due to GRPO's extensive exploration of the search space for selecting the correct concept labels, leading to sparse rewards and challenges in convergence.

In summary, SRJudge consistently outperforms alternatives on all datasets, underscoring its effectiveness in enhancing LLMs with selective reasoning through reinforcement learning. Consequently, this facilitates assigning the correct knowledge concept from numerous fine-grained candidates.

4.3 Ablation Study

Table 4 presents the results of our ablation studies to verify the effectiveness of the components and their various combinations in SRJudge. Overall, SRJudge consistently achieves the best performance across all ablation counterparts, demonstrating that each component contributes to the effectiveness of the proposed framework. Specifically, the Selector of SRJudge outperforms BERT-series models, as shown at the top of Table 3, because continued pretraining can further enhance the SLMs' ability to capture domain-specific contextual information and terminology distribution. Additionally, comparing baselines with "Selector", "Reasoner", and "Selector+Reasoner (**S+R**)" confirms the effectiveness of the LLM Reasoner in enhancing reasoning capabilities. After incorporating the Reasoner on top of the Selector, the F1 scores are improved by 2.02%, 0.96%, and 1.92% on the S_Math, S_Bio, and S_Phy datasets, respectively. These demonstrate that GRPO can effectively guide LLMs to align more closely with human annotation logic when reasoning about fine-grained concepts. Finally, integrating the Judger consistently enhances performance on all three datasets when comparing models using "Selector+Reasoner+Judger (**S+R+J**)" to those with "Selector+Reasoner (**S+R**)," highlighting the Judger's advantages in assessing the previous predictions from a global perspective. Thus, experimental results demonstrate that integrating all components is crucial, and a larger size of Judgers enhances the final performance of SRJudge.

Table 5 presents further ablation of the proposed dynamic position reward. SRJudge with position rewards consistently outperforms one without position rewards, which shows that introducing position rewards to the LLM Reasoner can prevent it from overfitting to the Selector's outputs. This not only improves the tagging performance by encouraging exploration of other candidate positions but also accelerates model convergence. Furthermore, the comparison between these two rewards demonstrates that dynamically adjusting rewards across positions and training steps yields additional improvements. This is because, as training progresses, the model tends to become more conservative, and the dynamic reward helps sustain active exploration and improve RL efficiency. Additionally, to investigate the impact of the pruning rate, we evaluate annotation performance and training time across different pruning-rate settings in Stage 2 (RL),

Model	S_Math				S_Bio				S_Phy			
	ACC	P	R	F1	ACC	P	R	F1	ACC	P	R	F1
Chinese-BERT (Selector, **S**)	0.7412	0.7503	0.7386	0.7289	0.7337	0.7045	0.6865	0.6829	0.6752	0.6748	0.6519	0.6437
Qwen2.5-1.5B-Instruct (Reasoner, **R**)	0.6454	0.6045	0.6057	0.5785	0.5879	0.4704	0.5468	0.4952	0.5548	0.4930	0.4705	0.4539
Qwen3-14B (Judger, $\mathbf{J}_1$)	0.5913	0.6161	0.5694	0.5488	0.5508	0.5182	0.5099	0.4883	0.3791	0.3831	0.3565	0.3373
Qwen3-32B (Judger, $\mathbf{J}_2$)	0.6093	0.6179	0.5913	0.5660	0.5688	0.5210	0.5251	0.4940	0.4018	0.4181	0.3952	0.3705
S + R	0.7649	0.7738	0.7573	0.7491	0.7417	0.7128	0.6986	0.6925	0.6871	0.6779	0.6667	0.6629
S + R + $\mathbf{J}_1$	0.7739	0.7760	0.7719	0.7592	0.7427	0.7146	0.7003	0.6946	0.6887	0.6779	0.6681	0.6634
S + R + $\mathbf{J}_2$	**0.7748**	**0.7772**	**0.7721**	**0.7602**	**0.7437**	**0.7285**	**0.7029**	**0.6987**	**0.6920**	**0.6791**	**0.6702**	**0.6643**

Table 4: Ablation Study of SRJudge. We provide additional ablation results of SRJudge variants using different Judgers.

Position reward	S_Math		S_Bio		S_Phy	
	ACC	F1	ACC	F1	ACC	F1
No	0.7559	0.7372	0.7397	0.6862	0.6806	0.6528
Static (Ours)	0.7608	0.7451	0.7387	0.6888	0.6853	0.6619
Dynamic (Ours)	**0.7649**	**0.7491**	**0.7417**	**0.6925**	**0.6871**	**0.6629**

Table 5: Ablation study on position reward strategies in RL. Static position reward sets $R_{\text{position}}(o_p) = 3$, whereas Dynamic position reward follows the standard computation in Equation (5).

Pruning rate	S_Math		S_Bio		S_Phy	
	F1	Time (h)	F1	Time (h)	F1	Time (h)
0.00	0.7404	14.1	0.6821	10.4	0.6603	24.3
0.25	0.7481	10.4	0.6912	6.7	0.6612	18.7
0.50	**0.7491**	7.2	**0.6925**	5.9	**0.6629**	13.4
0.75	0.7408	5.5	0.6902	4.2	0.6607	9.1

Table 6: Performance for RL using different pruning rates

as shown in Table 6. As the pruning rate increases, F1 scores exhibit a trend of initial improvement followed by a decline (while exceeding the no-pruning setting). Meanwhile, training time consistently decreases as the pruning rate increases. Overall, introducing the pruning mechanism not only improves macro F1 but also significantly enhances training efficiency. This is because, during the interaction between the agent and the environment, low-quality completions with positive but marginal advantages are effectively filtered out, reducing the agent's exposure to noisy feedback and the number of gradient updates, and accelerating model convergence.

4.4 Analysis on Hard-to-Distinguish Samples

To evaluate the effectiveness of SRJudge on hard-to-distinguish samples, we use the Selector's F1 score as a proxy for sample difficulty [Yang *et al.*, 2025], as illustrated in Figure 3. Note that if the F1 score for the knowledge concept category predicted by the Selector falls below 0.5, that category is deemed "difficult to distinguish". Using this criterion, we set five difficulty intervals as: [0, 0.1], [0.1, 0.2], [0.2, 0.3], [0.3, 0.4], and [0.4, 0.5]. For each difficulty interval, we evaluate the incremental effects brought by incorporating SRJudge. Experimental results show that SRJudge consistently achieves performance improvements across most difficulty intervals. These gains mainly arise from the enhanced

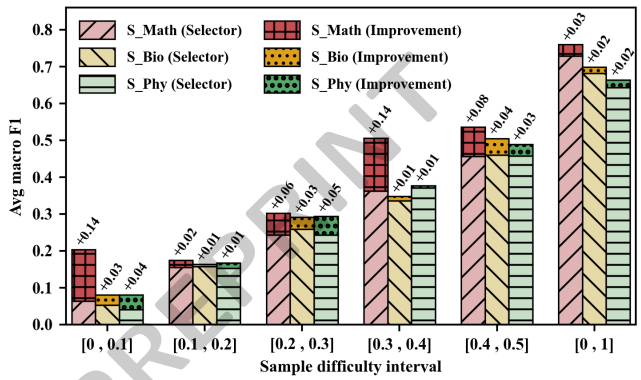

Figure 3: Performance of SRJudge on sample difficulty intervals.

reasoning and semantic understanding capabilities of the Reasoner trained via reinforcement learning, while the Judger further improves overall performance by integrating global contextual information and jointly evaluating the outputs of the Selector and the Reasoner. Notably, the performance gains of SRJudge on the higher-grade datasets S_Bio and S_Phy are relatively limited compared to those on S_Math, primarily because the knowledge concepts in these disciplines are more complex and interrelated, which hinders accurate annotation. Overall, SRJudge achieves average F1 improvements of 8.84% on S_Math, 2.31% on S_Bio, and 3.24% on S_Phy, demonstrating the clear effectiveness of SRJudge in improving knowledge concept prediction for hard-to-distinguish samples across different disciplines.

5 Conclusion

This paper presents a novel three-stage SRJudge framework that enhances LLMs with selective reasoning via reinforcement learning, enabling selection from numerous candidates. This method effectively integrates the reasoning strengths of LLMs with the efficient selection capabilities of small language models. To the best of our knowledge, this is the first work to customize RL to improve LLM reasoning over fine-grained knowledge concepts. Additionally, we construct two high-quality datasets to support follow-up research herein. Experimental results across datasets demonstrate the effectiveness of SRJudge for knowledge concept tagging, offering a streamlined framework that empowers LLMs with selective reasoning to distinguish fine-grained knowledge concepts.

Ethical Statement

There are no ethical issues.

Acknowledgments

This work is supported by the research project funded by Guangdong Basic and Applied Basic Research Foundation (2026A1515011829, 2024A1515140144), the Fundamental Research Funds for the Central Universities (21624325, 21624338), Ministry of Education of the People's Republic of China Humanities and Social Sciences Youth Foundation (24YJC890034), the Key Laboratory of Smart Education of Guangdong Higher Education Institute, Jinan University (2022LSYS003). This work is also partially supported by NSFC (62377028,62077028), Science and Technology Planning Project of Guangzhou Development District (2023GH01),"Master Mentor Plan" of Jinan University (YDXS2501), the Fundamental Research Funds for the Central Universities (21625102), the teaching reform research projects of Jinan University (JG2026030).

References

[Achiam *et al.*, 2023] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint:2303.08774*, 2023.

[Bai *et al.*, 2023] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint:2309.16609*, 2023.

[Cui *et al.*, 2024] Chaoran Cui, Yumo Yao, Chunyun Zhang, Hebo Ma, Yuling Ma, Zhaochun Ren, Chen Zhang, and James Ko. Dgekt: A dual graph ensemble learning method for knowledge tracing. *ACM Transactions on Information Systems*, 42(3):1–24, 2024.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 4171–4186, 2019.

[Ding *et al.*, 2025a] Ziqi Ding, Xiaolu Wang, Yuzhuo Wu, Guitao Cao, and Liangyu Chen. Tagging knowledge concepts for math problems based on multi-label text classification. *Expert Systems with Applications*, 267:126232, 2025.

[Ding *et al.*, 2025b] Ziqi Ding, Xiaolu Wang, Yuzhuo Wu, Guitao Cao, and Liangyu Chen. Tagging knowledge concepts for math problems based on multi-label text classification. *Expert Systems with Applications*, 267:126232, 2025.

[Ghosh *et al.*, 2024] Sreyan Ghosh, Chandra Kiran Reddy Evuru, Sonal Kumar, Deepali Aneja, Zeyu Jin, Ramani Duraiswami, Dinesh Manocha, et al. A closer look at the limitations of instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

[Gu, 2023] Qiuhan Gu. Llm-based code generation method for golang compiler testing. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 2201–2203, 2023.

[Guo *et al.*, 2025] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint:2501.12948*, 2025.

[Huang *et al.*, 2023] Tao Huang, Shengze Hu, Huali Yang, Jing Geng, Sannyuya Liu, Hao Zhang, and Zongkai Yang. Pqsct: Pseudo-siamese bert for concept tagging with both questions and solutions. *IEEE Transactions on Learning Technologies*, 16(5):831–846, 2023.

[Kullback and Leibler, 1951] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[Li *et al.*, 2022] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S Yu, and Lifang He. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology*, 13(2):1–41, 2022.

[Li *et al.*, 2024] Shiyu Li, Yang Tang, Shizhe Chen, and Xi Chen. Conan-embedding: General text embedding with more and better negative samples. *arXiv preprint:2408.15710*, 2024.

[Li *et al.*, 2025] Hang Li, Tianlong Xu, Ethan Chang, and Qingsong Wen. Knowledge tagging with large language model based multi-agent system. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, pages 28775–28782, 2025.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint:1907.11692*, 2019.

[Liu *et al.*, 2024] Yonghao Liu, Lan Huang, Fausto Giunchiglia, Xiaoyue Feng, and Renchu Guan. Improved graph contrastive learning for short text classification. In *Proceedings of the 38th AAAI conference on artificial intelligence*, pages 18716–18724, 2024.

[Moore *et al.*, 2024] Steven Moore, Robin Schmucker, Tom Mitchell, and John Stamper. Automated generation and tagging of knowledge components from multiple-choice questions. In *Proceedings of the 7th ACM conference on learning@ scale*, pages 122–133, 2024.

[Ozyurt *et al.*, 2024] Yilmazcan Ozyurt, Stefan Feuerriegel, and Mrinmaya Sachan. Automated knowledge concept annotation and question representation learning for knowledge tracing. *arXiv preprint:2410.01727*, 2024.

[Ozyurt *et al.*, 2025] Yilmazcan Ozyurt, Tunaberk Almaci, Stefan Feuerriegel, and Mrinmaya Sachan. Personalized

exercise recommendation with semantically-grounded knowledge tracing. In *Advances in Neural Information Processing Systems*, 2025.

[Saha and Schneider, 2013] Avigit K Saha and Kevin A Schneider. A discriminative model approach for suggesting tags automatically for stack overflow questions. In *2013 10th Working Conference on Mining Software Repositories*, pages 73–76. IEEE, 2013.

[Shao *et al.*, 2024] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint:2402.03300*, 2024.

[Sun *et al.*, 2018] Bo Sun, Yunzong Zhu, Yongkang Xiao, Rong Xiao, and Yungang Wei. Automatic question tagging with deep neural networks. *IEEE Transactions on Learning Technologies*, 12(1):29–43, 2018.

[Thirunavukarasu *et al.*, 2023] Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. Large language models in medicine. *Nature medicine*, 29(8):1930–1940, 2023.

[Touvron *et al.*, 2023] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint:2302.13971*, 2023.

[Wang *et al.*, 2024] Shanshan Wang, Zhen Zeng, Xun Yang, Ke Xu, and Xingyi Zhang. Boosting neural cognitive diagnosis with student's affective state modeling. In *Proceedings of the 38th AAAI conference on artificial intelligence*, pages 620–627, 2024.

[Xia *et al.*, 2025] Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. Evaluating mathematical reasoning beyond accuracy. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*, pages 27723–27730, 2025.

[Yang *et al.*, 2022] Jiuding Yang, Weidong Guo, Bang Liu, Yakun Yu, Chaoyue Wang, Jinwen Luo, Linglong Kong, Di Niu, and Zhen Wen. Tag: Toward accurate social media content tagging with a concept graph. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4332–4341, 2022.

[Yang *et al.*, 2025] Zhiwei Yang, Jiahua Yang, Longtao Wang, Rongxin Huo, Huiru Lin, Yuxuan Zhou, and Weiqi Luo. Local-global cascaded ensemble learning on hybrid experts for knowledge concept tagging. In *Proceedings of the 26th International Conference on Artificial Intelligence in Education*, pages 408–421. Springer, 2025.

[Zhang *et al.*, 2021] Xiao Zhang, Meng Liu, Jianhua Yin, Zhaochun Ren, and Liqiang Nie. Question tagging via graph-guided ranking. *ACM Transactions on Information Systems*, 40(1):1–23, 2021.

[Zhang *et al.*, 2025] Yazhou Zhang, Mengyao Wang, Qiuchi Li, Prayag Tiwari, and Jing Qin. Pushing the limit of llm capacity for text classification. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 1524–1528, 2025.