2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

CAMS: Collaborative Small-Parameter Large Language Models for Educational QA Grading

Gangliang Li[1], Dacheng Xu[1], Xiaodong Huang[1], Chengfeng Chen[1] and Shouqiang Liu[*1]

[1] School of Artificial Intelligence, South China Normal University, Foshan 528225, China
2023025171@m.scnu.edu.cn

Abstract. Automated grading has become a crucial component of smart education, involving various complex Natural Language Processing (NLP) tasks, including text representation, similarity evaluation, and classification. Although large language models (LLMs) show great promise in improving grading accuracy and consistency, their high computational costs and data privacy concerns limit widespread adoption. This study introduces CAMS, an automated grading system based on smaller LLMs that enhances the grading process through model collaboration and chain-of- thought (CoT)-guided prompt templates. CAMS offers an efficient, locally deployable, and sustainable solution. By integrating Yi-1.5-9B into the proposed collaborative CAMS system and deploying it locally, the system achieved a grading score of 0.8511 and an overall score of 0.8148, demonstrating improvements of 0.1865 and 0.1732, respectively, compared to the standalone use of Yi-1.5-9B. Furthermore, the performance of CAMS approaches that of larger-scale model APIs such as GPT-3.5- Turbo (score = 0.8457).

Keywords: large language models · automatic grading · prompt engineering · natural language processing.

1 Introduction

Automated grading plays a crucial role in smart education, with its significance increasing alongside the digitalization of learning environments [1]. This process involves several complex Natural Language Processing (NLP) tasks, such as text representation, similarity assessment, and classification, while also ad- dressing educationally specific challenges, including domain-specific knowledge, answer diversity, and various types of errors. Grading methodologies have gradually evolved from rule-based and statistical models to more sophisticated techniques, such as neural networks, meta-learning, and pre-trained models. Current research primarily focuses on how to improve the accuracy and generalizability of automated grading systems.

Large language models (LLMs), with advanced capabilities in semantic understanding and text generation, demonstrate strong performance in a variety of NLP tasks. In the context of automated grading, LLMs contribute to improved accuracy and

consistency while offering interpretability for complex linguistic analysis. However, despite high precision (e.g., GPT-4), their adoption remains limited by high computational costs and concerns over data privacy [2]. Local deployment reduces API costs but requires high-performance hardware, creating barriers. Thus, the local deployment of smaller LLMs presents a critical challenge in achieving cost-effective and sustainable grading solutions [3].

This research aims to develop an automated grading system for knowledge-based responses through the collaborative capabilities of large language models. The proposed system decomposes complex grading tasks into smaller sub-tasks and utilizes prompt templates to facilitate collaboration, thereby improving the model's comprehension and consistency in evaluating user responses. The effectiveness of this approach is demonstrated through specific examples in **Fig. 1**. Experiments will be conducted with 6–9 billion parameters to investigate the feasibility of collaboration among smaller models, with the goal of achieving grading performance comparable to that of large-scale online LLM APIs.

This work makes the following contributions:

1. We conduct a comparative evaluation of automated grading performance across 11 large language model APIs and 6 locally deployed models, aiming to assess their effectiveness under low-parameter conditions.
2. We propose a locally deployable collaboration system, CAMS, based on 6-9B large language models. CAMS achieves automated automatic grading performance close to that of large parameter models such as GPT-3.5-Turbo and GLM-3-Turbo. The system provides efficient grading functionality, making it suitable for environments with limited computational resources.

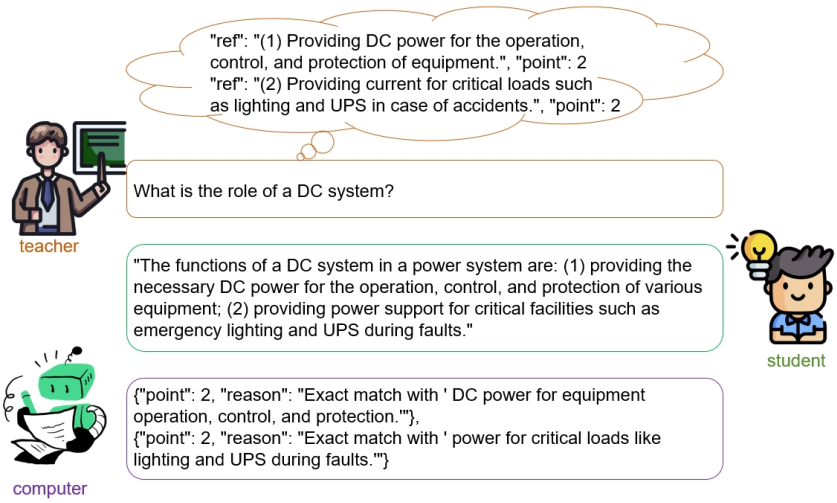

Fig. 1. Example of automated grading using CAMS: The teacher poses questions along with reference answers; students submit their responses, which are then evaluated and scored by the system.

2 Related Works

2.1 Automated Grading Systems and Challenges

Automated grading is a crucial tool for assessing learning outcomes, addressing the time and cost challenges of manual grading [4]. With the rise of online education and large-scale assessments, ensuring the accuracy and fairness of automated grading has become a key research focus. Several methods have been developed to support this objective, including:

- Concept Mapping: Identifies underlying concepts in student answers and uses rule-based algorithms to assess alignment with reference answers [5]. Partial credit is given based on concept similarity.
- Information Extraction: Matches patterns from resources like textbooks or reference answers to student responses. Teodora's system uses Latent Semantic Analysis (LSA) to assess the semantic similarity between student answers and reference materials [6].
- Corpus-Based Methods: Analyzes large corpora to identify key terms and compute semantic distances between student responses and reference answers [7]. Techniques like LSA and Singular Value Decomposition (SVD) generate a semantic space for distance-based evaluation.
- Machine Learning (ML) Methods: Uses ML models with handcrafted features from NLP techniques to grade student responses. Mohler demonstrated significant improvements in grading performance using machine learning [8].

2.2 Large Language Models in Education

LLMs, such as OpenAI's GPT-3 and Google's BERT, are advanced AI systems that have transformed NLP tasks with their exceptional capabilities in language understanding and generation. These models excel in producing fluent text, machine translation, summarization, and question answering by analyzing complex syntactic and semantic patterns. LLMs are revolutionizing NLP applications by enhancing human-machine interactions, automating content creation, and improving intelligent customer service [9]. In the educational domain, LLMs demonstrate remarkable abilities in semantic generation, particularly in tasks such as automated grading. They generate contextually relevant content and improve grading accuracy by overcoming the limitations of traditional methods, such as the handling of synonyms with synonyms, thus boosting efficiency and accuracy [10].

Transformers in LLMs use attention mechanisms to capture relationships between words, thereby enhancing the efficiency and accuracy of NLP tasks [11]. This architecture consists of an encoder that extracts linguistic patterns and a decoder that generates coherent text using multi-head attention. SoftMax layer is used to select the most

probable next word in the sequence. This design enables LLMs to produce contextually relevant text, enhancing NLP capabilities.

3 Method

As shown in **Fig. 2**, traditional grading methods rely on small-scale models that process inputs in a single-pass manner. These methods struggle to handle complex semantic structures and scoring logic, resulting in reduced accuracy and consistency. Common issues include score-reasoning mismatch; misunderstanding the requirements; and incomplete grading.

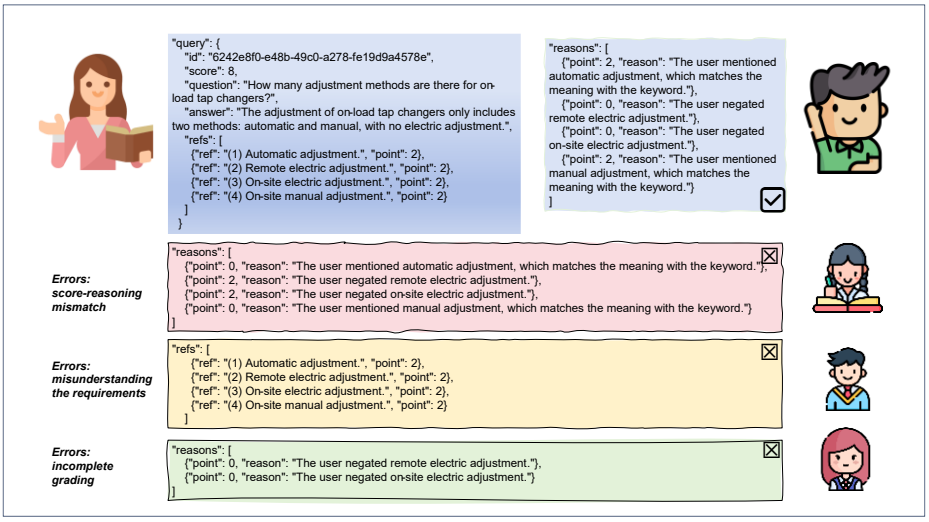

Fig. 2. Common Errors: score-reasoning mismatch; misunderstanding the requirements; incomplete grading.

Experimental results have demonstrated that the use of multiple LLM modules for collaborative tasks can effectively address functional gaps and significantly enhance the grading performance of user responses. Based on these findings, this study proposes a system called the Collaborative Attention-based Multilayer System (CAMS). Within the system, each LLM module is assigned a specific functional prompt. To minimize local deployment costs, identical LLM models can be utilized across modules; however, to improve overall functionality, different specialized LLM models may be employed.

3.1 Prompt Design for Individual Modules:

The prompt design for individual modules integrates several advanced techniques, including Chain of Thought (CoT), System2Attention, and Few-shot learning. The combined application of these methods aims to address challenges related to accuracy,

consistency, and information transmission, which LLM often encounter when handling complex tasks.

CoT [12] architectures decompose reasoning into structured intermediate steps with transitional continuity verification, contrasting with monolithic answer- generation paradigms. Dual mechanisms ensure: (I) cognitive coherence through step transition constraints [13], and (II) transparent deduction rule instantiation for each computational phase [14]. This dual-output framework simultaneously generates final predictions and step validity attestations, intrinsically resolving syntactic-semantic discordance commonly found in traditional single-pass systems.

Few-shot prompting improves output consistency in resource-constrained LLMs [15] through exemplar-driven pattern induction, where strategically retrained QA pairs in conversational history create implicit template scaffolding. By conditioning model responses on precedent interaction patterns through in-context learning, this approach achieves style-consistent generation via cognitive analogy transfer. This effectively bypasses architectural constraints through dynamic demonstration grounding rather than parametric expansion.

System2Attention [16] mitigates stochastic instability in resource-constrained LLMs by referencing Few-shot historical chat records to extract target information and output JSON output. This mechanism addresses parameterization-induced randomness through attention-based saliency mapping [17], effectively pruning irrelevant outputs while preserving semantic integrity [18]. The JSON output schema ensures the removal of unnecessary data, maintaining focus on relevant outputs. It enforces structured data integrity in inter-module communication, addressing format inconsistencies inherent in traditional methods. Through its recursive syntax validation and standardized serialization, it guarantees type-safe encapsulation of hierarchical dataflows, eliminating parsing ambiguities. This isomorphic representation framework fosters cross-component interpretability, thereby enhancing collaborative stability, ensuring all modules can consistently interpret and process data.

3.2 Collaborative Module Function Design:

Building upon individual modules, the collaboration between different LLM components can more effectively accomplish tasks. Given the limited capabilities of the small LLM model used in this study, which cannot autonomously design collaborative workflows, three core prompts were specifically developed to address the issue of insufficient output accuracy in smaller models when handling complex tasks. These prompts focus on keyword extraction, answer-keyword matching, and scoring based on the matching results. The design prompts for the three modules are as follows.

The keyword extraction module adopts a two-stage processing mechanism. Initially, it employs refined extraction techniques to identify 2-4 core keywords corresponding to each scoring point from the reference answer, ensuring comprehensive coverage of the typical characteristics of the knowledge elements. Subsequently, it applies intelligent filtering through semantic matching to automatically eliminate redundant terms that overlap with the question, effectively extracting the keywords relevant to answering the question. The system ultimately outputs structured data in compliance with

JSON standards, accurately presenting the key information of the reference points independent of the question content.

Keyword Module Prompt	**Example**
Role: Professional semantic mining AI that extracts core scoring elements from reference answers with exam-level precision. **Workflow:** 1) **Refine**: Extract 2--4 critical keywords per point. 2) **Filter**: Auto-remove redundant terms matching the question. **Output Format:** {"key raw": "<Unique keywords from 'ref', excluding terms in 'question'>"}	**Input JSON:** {"id": "3a3ca9c4-8773-41bb-baec-da5bb8f7a24e", "question": "Under what circumstances can the power supply to relevant equipment be disconnected without permission?", "answer": "If the weather is too hot or if one is leaving the work site for lunch, then it may be considered to disconnect the power.", "refs": [{"ref": "(1) In the event of a personal electric shock accident, the power to relevant equipment can be disconnected without permission to rescue the person affected by the shock.", "point": 2}]} **Reference content:** The reference content is: "In the event of a personal electric shock accident, the power to relevant equipment can be disconnected without permission to rescue the person affected by the shock." 1) **Refine**: 'personal electric shock accident', 'rescue the person affected by the shock', 'relevant equipment' 2) **Filter**: Repeated words include 'relevant equipment' **Final output JSON**: "Key raw": "personal electric shock accident; rescue the person affected by the shock;"

The Answer-Keyword matching module is specifically designed for in-depth matching analysis between student answers and scoring keywords, based on a layered verification mechanism of "explicit-implicit-conflict." Initially, it checks whether the student answer explicitly contains the original words or synonyms of the scoring keywords. Subsequently, it utilizes semantic inference to identify associated expressions of attribute features, achieving implicit feature mapping (e.g., equating "quiet operation" with "no abnormal sounds"). Finally, it applies a conflict detection algorithm to identify and analyze contradictory expressions (e.g., the mutual exclusivity between "clean" and "stained,") and generates a structured conflict report based on the identified discrepancies.

The Scoring module is an intelligent scoring decision system based on multidimensional feature analysis, which facilitates a graded quantification assessment of students' mastery of knowledge. It innovatively constructs a dynamic weight distribution model, assigning a baseline weight of (0.5-1) based on the importance of each knowledge point. Through a three-level evaluation mechanism, it precisely scores by first

establishing a baseline scoring framework with categories of (complete match (1) - partial match (0.5) - conflict/missing (0)); secondly it introduces a semantic feature dynamic adjustment mechanism, applying a weight decay of 0.2 to redundant expressions and a score enhancement of 0.2 for the accurate use of professional terminology; and finally, it calculates the final score by multiplying multiple dimensional coefficients.

Matching Module Prompt	**Example**
Role: As a Professional Core Validator, perform 3-tier semantic verification (beyond keyword matching) using contextual logic validation. **Validation Process:** 1) **Explicit verification:** Verify exact/synonym matches (using power industry term database). 2) **Implicit verification:** Detect attribute-related expressions (e.g., “runs quietly”” no abnormal noise”). 3) **Contradiction verification:** Flag conflicting statements (e.g., ``clean" vs. ``stains present"). **Feedback Rules:** **Mentioned**: Mark evidence location. **Implicitly Mentioned**: Annotate contextual linkage grounds. **Conflicting**: Explain exact contradiction. **Output Format:** {"reason": "<mentioned content; Implicitly mentioned content; conflicting content>"}	**Input JSON:** {“key raw": "personal electric shock accident; rescue the person affected by the shock;"} **Answer:** "If the weather is too hot or if one is leaving the work site for lunch, then it may be considered to disconnect the power." 1) **Explicit verification:** Keywords not directly or synonymously mentioned. 2) **Implicit verification:** Expressions such as "too hot", "lunch" are not related to safety/emergency. 3) **Contradiction verification:** Disconnecting power due to lunch/weather contradicts the premise of the emergency context. **Final output JSON**: "reason": "personal electric shock accident not mentioned; rescue the person affected by the shock not mentioned"

The score is mapped as follows: 0 for "Not Mentioned" (0-0.7), 1 for "Partial Mentioned" (0.7-1.3), and 2 for "Full Mentioned" (1.3-2).

4 Experiments

4.1 Dataset

Due to the lack of relevant datasets in this field, this study will utilize the GPT-4o online API to simulate diverse student responses to questions from a specific question bank. Each question contains 1 to 6 key points, covering a range of content from simple to complex and highly specialized. Specifically, the simulation will generate five types of answers to comprehensively evaluate the effectiveness of the automatic scoring system:

- Perfect Answer: A comprehensive and accurate response, fully aligned with the reference answer. This represents the ideal case, where the model should recognize the highest-quality response for scoring.

Scoring Module Prompt	**Example**
Role: As the Scoring Decision Hub, dynamically construct a scoring model that reflects students' depth of understanding through gradient scoring. **Scoring Process:** 1) **Weights**: Rank keywords by importance (0.5--1.0). 2) **Baseline Scores**: Mentioned: Weight 1 Implicitly mentioned: Weight 0.5 Contradiction/not mentioned: 0 3) **Adjustments:** Repeated mentions: 0.8 (anti-redundancy) Technical terms: 1.2 (only mentioned) 4) **Final Calculation**: **Total Score**=Weight * Adjustment Factor **Actual Score** = Weight * Base Score * Adjustment Factor/Total Score *Standard score **Output Format:** {"get point": " Actual Score"}	**Input JSON:** reason: "personal electric shock accident not mentioned; rescue the person affected by the shock not mentioned". point: 2 **Scoring Process:** 1) **Weights:** Personal electric shock accident: 1.0 Rescue shocked person: 0.8 2) **Baseline Scores**: personal electric shock accident not mentioned: 0 rescue the person affected by the shock not mentioned: 0 3) **Adjustments:** No repetition: 1 Professional terms: 1 4) **Final Calculation:** Total Score: (1.0 + 0.8) * 1 * 1= 1.8 Actual Score: [(1.0 * 0 * 1 * 1) + (0.8 * 0 * 1 * 1)] / 1.8 * 2 = 0 4) **Final output JSON**: "get point": "0"

- Paraphrased Answer: The student restates the key points with slight variations in wording. This tests the model's ability to recognize semantic equivalence despite different expressions or phrasing.
- Partial Answer: A response that covers only some of the key points, sometimes using near-synonyms. This tests the model's ability to handle incomplete answers and assess missing key information.
- Incorrect Answer: A deliberately incorrect response, sometimes containing factual errors or contradictions. This helps evaluate the model's ability to identify inaccuracies or fallacies in the answer.
- Irrelevant Answer: A response that does not address the question at all. This tests the model's ability to identify whether the content is on-topic and ensures that irrelevant responses are appropriately scored.

Manual review is performed by three evaluators with expertise in the dataset's domain (including corrective verification of samples with factual/logical errors) to ensure accuracy and reliability, thereby facilitating subsequent research. Since the model has not previously encountered a dataset from this domain, this task is categorized as zero-shot transfer. Given the diversity of student responses in the dataset and the need for subsequent manual evaluation of the model's grading performance, a total of 1390 samples were selected. This dataset is sufficient for assessing the model's ability to handle

different types of answers and their associated complexities, while also enabling the manual evaluation of the model's grading effectiveness.

Our experiments employed an NVIDIA A100 GPU with the VLLM inference framework. The model was loaded from a local path and configured with temperature = 0.3 and max_tokens = 3000 sampling parameters. During deployment, the system consumed approximately 24 GB of GPU memory.

4.2 Evaluation Method

The input for automated grading consists of an ID, question, answer, and a reference set with scoring points and their corresponding scores. The output includes the reasoning and final score. To address potential inaccuracies in manual evaluations, this study proposes a combined manual and automated evaluation method (**Fig. 3**) using a weighted summation strategy. The method comprises three components: Automated Reasoning Quality, Manual Reasoning Quality, and Automated Scoring Quality Evaluation. The evaluation metrics are defined as follows:

Automated Reasoning Quality Evaluation: BERT Score uses cosine similarity between words in the generated and reference texts. The evaluation metrics include Precision, Recall, and F1-score, defined as follows:

$$\text{Precision} = \frac{\sum_{w \in \text{generated}} \text{sim}(w, \hat{w})}{\sum_{w \in \text{generated}} 1} \tag{1}$$

$$\text{Recall} = \frac{\sum_{w \in \text{reference}} \text{sim}(w, \hat{w})}{\sum_{w \in \text{reference}} 1} \tag{2}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

The automatic evaluation task often encounters cases where sentences like "Tomorrow is Friday" and "Tomorrow is not Friday" appear highly similar on the surface but are semantically contradictory in reasoning. Traditional metrics, such as the F1-score, may fail to accurately assess reasoning validity in such cases. In response, a new evaluation metric, F1 NLI, is introduced. The F1 NLI metric extends the traditional F1 score by evaluating the relationship between the reference and candidate using a natural language inference (NLI) model. Specifically, the F1 NLI score is computed using the MoritzLaurer/mDeBERTa-v3-base-mnli-xnli model. If the reference and candidate are found to contradict, the F1 NLI score is set to 0. Otherwise, it is equal to the traditional F1 score. This metric adds an additional layer of logical consistency to the evaluation process, improving the accuracy of reasoning evaluation.

Manual Evaluation of Reasoning Quality: In manual evaluation, three independent reviewers assess the quality of reasoning by comparing the generated rationale to the reference point. The reviewers rate the rationale on a scale from 0 to 4, where a score of 4 indicates high consistency with the reference meaning and content, 3 indicates insufficient detail, 1 represents partial matching, and 0 means the rationale contradicts the reference. The final quality score is obtained by aggregating these individual scores

according to a predefined weighted scheme, which is then normalized to ensure consistency.

Automated Scoring Quality Evaluation: For reasoning quality evaluation scores (F1 NLI and Human NLI), which range from 0 to 1, it is necessary to normalize the automated scoring quality evaluation scores. In the normalization process, the difference between the predicted and true values is calculated, and then adjusted using a squared difference method. This procedure ensures that scores from different evaluation metrics are standardized and comparable, promoting consistency across assessment methods.

$$\text{score} = \frac{1}{1 + (Predicted_{\text{score}} - True_{\text{score}})^2} \tag{4}$$

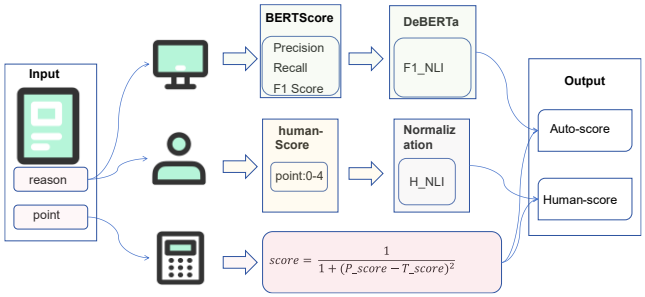

Fig. 3. Evaluation process: Three types of evaluations are conducted on reasoning and scoring points, ultimately yielding both the automatic and human evaluation results.

The experimental results established the significant superiority of F1 NLI metrics over conventional F1 metrics across evaluation dimensions. Statistical analysis revealed that F1 NLI achieved a 3.2% higher Pearson correlation with human evaluations (0.9874 vs. 0.9598) and a 1.3% improvement in score similarity alignment (0.9979 vs. 0.9847). Both differences were statistically significant ($p<0.001$), affirming F1 NLI's superior ability to replicate human judgment patterns.

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})\,(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{5}$$

The coefficient r ranges from -1 to 1, indicating the strength and direction of the linear relationship.

4.3 Effectiveness of Collaborative Attention-based Multilayer System

To evaluate the effectiveness of CAMS system, six local models were evaluated on grading tasks. Their performance, with and without CAMS implementation, was benchmarked against four major LLM APIs (GPT-3.5-Turbo, GLM-3-Turbo, GLM-4, GPT-4o) using identical datasets.

Table 1 demonstrates two findings. First, CAMS integration consistently enhances all four metrics (Score, Auto, Human, and Average) across six locally deployed small models. The smallest model (chatglm3-6B-chat) exhibits a 320% improvement in

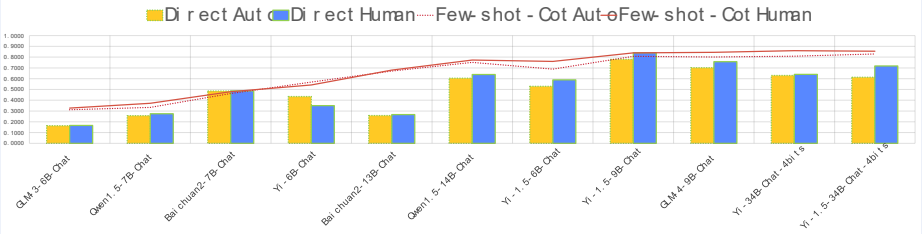

Fig. 4. Visualization of Automated vs. Manual Assessment: Yellow bars represent automatic evaluation with direct prompting, blue bars denote human evaluation, and the red line shows few-shot-CoT prompting. Solid lines indicate human evaluation, dashed lines indicate automatic evaluation.

Average score (0.1211 to 0.5084), while the largest model (LLaMA3.1-8B-Chat) achieves an 8.9% gain. Notably, the magnitude of improvement inversely correlates with model size: 6B-parameter Yi-1.5-6B-chat elevates its Human score by 26.5%, the 28.2% gain of the 9B-parameter variant, high- lighting CAMS's effectiveness in mitigating representation limitations in smaller models.

Second, CAMS-enhanced small models rival cloud-based large APIs. llama3.1- 8B-chat+CAMS (Avg: 0.8249) outperforms GPT-3.5-Turbo by 14.4% and approaches GLM-4 (0.8333). Strikingly, Yi-1.5-9B-chat+CAMS achieves a Human evaluation score of 0.8524, merely 0.96% below GPT-4o, while its Score metric (0.8511) exceeds GLM-3-Turbo by 9.2%. These findings challenge the conventional parameter-performance paradigm, showing collaborative attention enables models with ≤9 billion parameters to attain 80–95% of the accuracy of models with hundreds of billions of parameters.

These results substantiate CAMS's dual advantages: (I) multilayer attention synergy mitigates the capacity constraints of small models, yielding substantial improvements in Average scores; and (II) enhanced local models, such as LLaMA3.1-8B-Chat with CAMS, achieve near state-of-the-art performance, providing privacy-preserving alternatives for educational applications without compromising grading quality.

Fig. 6 reveals two critical patterns. The performance gains of CAMS across different question types exhibit significant disparities. The "Irrelevant" category demonstrates an average improvement of 27.5 percentage points (pp), with the maximum single-model gain reaching 45.98 pp. The "Perfect" category shows a 22.6 pp average enhancement (peaking at 42.98 pp for individual models), while the "Partial" category achieves only a 13.2 pp gain. A reverse correlation emerges between model capacity and enhancement efficacy: low-capacity models (GLM3- 6B/Qwen1.5-7B) attain maximum gains of 42.98 pp/47.55 pp in "Perfect" and "Paraphrased" categories; medium-capacity models (Yi-1.5 series) achieve ap- proximately 25 pp/19 pp improvements in "Irrelevant" and "Paraphrased" categories; high-capacity models (GLM4-9B/Llama3.1-8B) maintain double-digit gains in "Incorrect" and "Irrelevant" categories but exhibit negative gains in "Paraphrased".

CAMS' core advantage manifests through polar answer reinforcement: systematic performance elevation stems from enhanced robustness in "Perfect" category

Table 1. Performance evaluation of Collaborative Attention-based Multilayer System

Model	score	auto	human	avg
chatglm3-6B-chat	0.1414	0.1171	0.1251	0.1211
qwen1.5-7B-chat	0.2445	0.2357	0.2611	0.2484
glm-4-9B-chat	0.6131	0.5329	0.5923	0.5626
Yi-1.5-6B-chat	0.6125	0.5250	0.5893	0.5572
Yi-1.5-9B-chat	0.6646	0.6183	0.6649	0.6416
llama3.1-8B-chat	0.8020	0.7247	0.7905	0.7576
chatglm3-6B-chat+CAMS	0.5600	0.5126	0.5041	0.5084
qwen1.5-7B-chat+CAMS	0.6024	0.5385	0.5614	0.5500
glm-4-9B-chat+CAMS	0.7366	0.7141	0.7438	0.7290
Yi-1.5-6B-chat+CAMS	0.7581	0.6813	0.7456	0.7135
Yi-1.5-9B-chat+CAMS	**0.8511**	**0.7771**	**0.8524**	**0.8148**
llama3.1-8B-chat+CAMS	**0.8559**	**0.7886**	**0.8611**	**0.8249**
GPT-3.5-Turbo	0.8457	0.7052	0.7370	0.7211
GLM-3-Turbo	0.7795	0.7191	0.7539	0.7365
GLM-4	0.9030	0.8122	0.8544	0.8333
GPT-4o	0.9120	0.8245	0.8620	0.8433

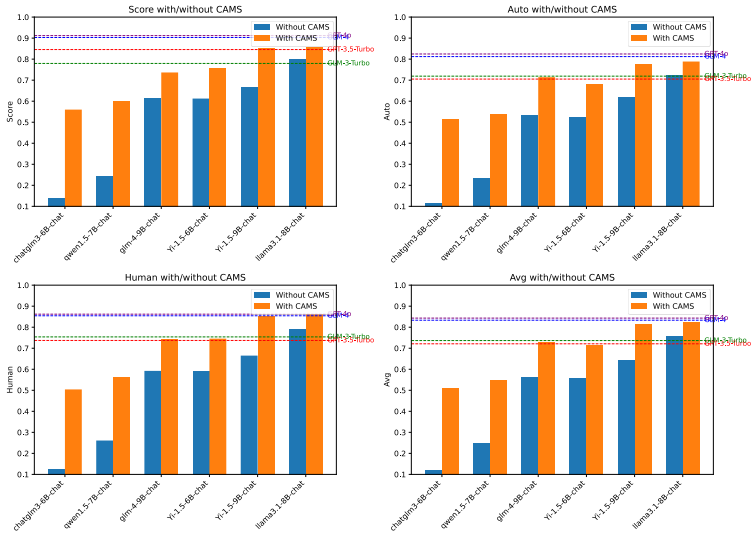

Fig. 5. Two-tier Comparison: (a) CAMS Efficacy in 6 Local LLMs; (b) Enhanced Local Models vs. Cloud APIs (GPT/GLM Series)

recognition (over 20 pp overall gain) and strengthened filtering capability against “Irrelevant" distractors (average 28 pp gain). Its marginal effects diminish with increasing

model capacity, showing most pronounced support for low-capacity models (e.g., Qwen1.5-7B achieves the maximum single-category gain of 47.55 pp in “Paraphrased"), while high-capacity models retain advantages only in specific categories (e.g., GLM4-9B’s 45.98 pp gain in "Irrelevant"). Three breakthrough cases (Qwen1.5-7B/GLM4-9B/GLM3-6B) collectively validate CAMS’ technical capability in precisely addressing Direct method’s weaknesses through targeted enhancement.

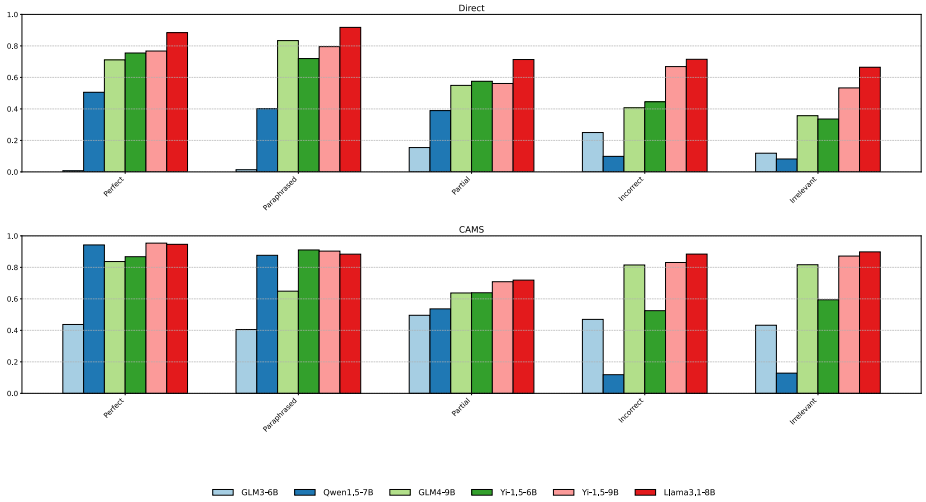

Fig. 6. Cross-Method Category-Specific Classification Performance Across Varied-Scale Language Models

4.4 Ablation Study

As shown in **Fig. 7**, assesses the performance of various models under two configurations:

Few-Shot Chain-of-Thought (FSC): All six local models exhibited consistent performance improvements when employing FSC prompting. The least performant model, ChatGLM3-6B-Chat, enhanced its Average metric from 0.1211 to 0.1372 (+13.3%), while the robust baseline, LLaMA3.1-8B-Chat, achieved a notable improvement (0.7576 to 0.8123, +7.2%). Medium-scale models demonstrated the most substantial gains: Qwen1.5-7B-Chat recorded a 67.4% increase in Average score (0.2484 to 0.4158) under standalone FSC, confirming the effectiveness of example-guided reasoning for tasks of moderate complexity Collaborative Module configurations.

Collaborative Module: The Collaborative Module demonstrated a "compensatory enhancement" effect, significantly improving the performance of the least effective model, ChatGLM3-6B-Chat, by 287% in Average score (0.1211 to 0.4688), while enhancing the high-performing GLM-4-9B-Chat by a more modest 26.4% (0.5626 to 0.7114). An anomalous observation was noted: LLaMA3.1-8B- Chat experienced performance degradation when using the Collaborative Module alone (Average: 0.7576 to 0.5733), indicating that its complex architecture may require synergistic integration with Few-Shot Chain-of-Thought prompting to facilitate effective knowledge transfer.

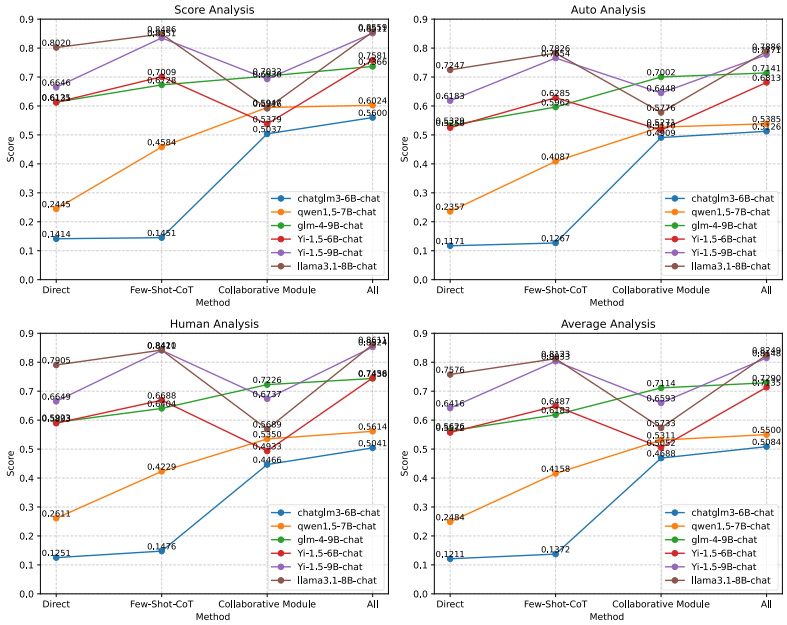

Fig. 7. Ablation analysis of model components (Direct, Few-Shot-CoT, Collaborative Module, All) across four evaluation metrics: Score, Auto, Human, and Average

A super additive synergy was observed: the combined gain for chatglm3-6B- chat's combined gain (320%) far exceeded the sum of individual improvements from FSC (13.3%) and CM (287%). For 9B-scale models, Yi-1.5-9B-chat's combined performance (0.8148) surpassed its best single-module result (FSC:0.8033), achieving a Human evaluation score of 0.8524, which was 26.5% higher than the CM-only configuration. Systemic superiority was significant, as the combined approach enabled all models to outperform their single-module peaks. For instance, Qwen1.5-7B-Chat's combined Avg (0.5500) outperformed its CM-only score (0.5311) by 3.6%. FSC establishes local reasoning pathways through exemplar guidance, while CM facilitates global knowledge distillation via cross-layer attention. Their spatiotemporal complementarity manifests as FSC addressing "how to reason" and CM focusing on "what to reason", forming a complete cognitive loop when combined.

In conclusion, the combination of FSC and CM significantly enhances model performance, particularly for smaller models. The experimental data demonstrates that the combined application of these two techniques significantly improves grading effectiveness, validating the effectiveness of the proposed approach.

5 Conclusion

This paper proposes CAMS, which leverages the synergy of multiple low-parameter LLM modules to address the limitations of traditional grading methods, such as limited processing capacity, shallow semantic understanding, and inconsistent scoring logic. In the system architecture, low-parameter LLMs collaborate across distinct modules to tackle complex tasks. While individual models may exhibit limited performance, their collaboration, supported by prompt engineering, leverages their strengths to improve overall system performance. Techniques such as CoT, System2Attention, and Few-Shot Learning further enhance inter-model communication, ensuring robust collaborative effectiveness. Experimental results demonstrate that the collaboration of multiple low-parameter LLMs effectively mitigates functional deficiencies, enhancing the accuracy and consistency of grading user responses.

In future work, we aim to: (I) investigate the dynamic collaboration capabilities of the system, analyzing its adaptability to varying input complexities and its impact on overall performance; (II) explore the automation of prompt generation and its influence on model reasoning and efficiency; and (III) assess the scalability of the approach for addressing more complex tasks and its applicability to a broader spectrum of natural language processing challenges. Additionally, we plan to explore the automation of prompt generation and its influence on model reasoning and efficiency. Furthermore, we will examine the scalability of the approach in handling more complex tasks and its application to a wider range of natural language processing challenges efficient and scalable.

References

1. Baidoo-Anu, D. and Ansah, L. O.: Education in the era of generative artificial intelligence (AI): Understanding the potential benefits of ChatGPT in promoting teaching and learning. Journal of AI 7(1), 52–62 (2023)
2. Tülübaş, T., Demirkol, M., Ozdemir, T. Y., Polat, H., Karakose, T. and Yirci, R.: An interview with ChatGPT on emergency remote teaching: A comparative analysis based on human–AI collaboration. Educational Process: International Journal 12(2), 93 (2023)
3. Roos, J., Kasapovic, A., Jansen, T., Kaczmarczyk, R., et al.: Artificial intelligence in medical education: comparative analysis of ChatGPT, Bing, and medical students in Germany. JMIR Medical Education 9(1), e46482 (2023)
4. Burrows, S., Gurevych, I. and Stein, B.: The eras and trends of automatic short answer grading. International Journal of Artificial Intelligence in Education 25, 60–117 (2015)
5. Leacock, C. and Chodorow, M.: C-rater: Automated scoring of short-answer questions. Computers and the Humanities 37, 389–405 (2003)
6. Mihajlov, T.: Automatic Student Answer Assessment using LSA. In: Proceedings of the Workshop on Computational Terminology in NLP and Translation Studies (ConTeNTS) Incorporating the 16th Workshop on Building and Using Comparable Corpora (BUCC), pp. 36–44 (2023)
7. Klein, R., Kyrilov, A. and Tokman, M.: Automated assessment of short free-text responses in computer science using latent semantic analysis. In: Proceedings of the 16th Annual Joint

Conference on Innovation and Technology in Computer Science Education, pp. 158–162 (2011)
8. Mohler, M., Bunescu, R. and Mihalcea, R.: Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 752–762 (2011)
9. Warudkar, S. and Jalit, R.: Unlocking the Potential of Generative AI in Large Language Models. In: 2024 Parul International Conference on Engineering and Technology (PICET), pp. 1–5 (2024)
10. Parsing, C.: Speech and language processing. Power Point Slides (2009)
11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems 30 (2017)
12. Zhao, H., Yilahun, H. and Hamdulla, A.: Pipeline chain-of-thought: A prompt method for large language model relation extraction. In: 2023 International Conference on Asian Language Processing (IALP), pp. 31–36 (2023)
13. Wang, Z., Zhou, B., Liu, W.: Atscot: Chain of thought for structuring anesthesia medical records. In: 2024 International Conference on New Trends in Computational Intelligence (NTCI). pp. 145–149. IEEE (2024)
14. Carlander, D., Okada, K., Engström, H., Kurabayashi, S.: Controlled chain of thought: Eliciting role-play understanding in llm through prompts. In: 2024 IEEE Conference on Games (CoG). pp. 1–4. IEEE (2024)
15. Weston, J. and Sukhbaatar, S.: System 2 attention (is something you might need too). arXiv preprint arXiv:2311.11829 (2023)
16. Cho, S., Seo, J., Jeong, S. and Park, J. C.: Improving zero-shot reader by reducing distractions from irrelevant documents in open-domain question answering. arXiv preprint arXiv:2310.17490 (2023)
17. Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., Weston, J.: Chain-of-verification reduces hallucination in large language models. arXiv preprint arXiv:2309.11495 (2023)
18. Giobergia, F., Koudounas, A., Baralis, E.: Large language models-aided literature reviews: A study on few-shot relevance classification. In: 2024 IEEE 18th International Conference on Application of Information and Communication Technologies (AICT). pp. 1–5. IEEE (2024)