DiffCL: Difference-Aware Contrastive Learning for Automatic Answer Grading with Multi-Level Semantic Modeling

Lei Chen[1], BoYu Gao[1,2], Zitao Liu[1], Tingjie Wan[1], Weiqi Luo[1],
[1]Guangdong Institute of Smart Education, Jinan University, Guangzhou, China,
[2]College of Cyber Security, Jinan University, Guangzhou, China,
Correspondence: bygao@jnu.edu.cn

Abstract

Automated Answer Grading (AAG) is a fundamental task in intelligent education, requiring accurate semantic understanding and reliable modeling of student deviations from reference answers. Despite recent progress, large language models (LLMs) remain insensitive to missing key concepts, exhibit unstable scoring scales, and lack structured scoring semantics in their representation space. To overcome these limitations, we propose a difference-aware AAG framework that integrates heuristic difference labeling with dual-contrastive learning. Semantic difference levels between student and reference answers are automatically inferred through similarity-based heuristics and injected into the model input as explicit prompts, enabling fine-grained perception of semantic deviations. In addition, an InfoNCE-based contrastive objective enforces representation consistency among samples with identical scores, while a hierarchical contrastive constraint guided by score gaps promotes structured separation across different scoring levels. Experiments on benchmark datasets, including SciEntsBank and Beetle, show that the proposed method consistently outperforms cross-entropy–based baselines in accuracy, weighted accuracy, and relevance metrics. Further analyses demonstrate improved robustness and generalization, even when applied to small-scale models. We have made all datasets and the corresponding code publicly accessible at: https://github.com/leibnizchen/DiffCL

1 Introduction

Automated Answer Grading (AAG) aims to provide objective and consistent grading of students' open-ended text answers and is one of the core technologies in intelligent teaching systems and online education platforms (Mohler and Mihalcea, 2009). Unlike multiple-choice question grading, short-answer grading requires the model not only to determine whether the answer is correct but also to understand whether the student's expression covers key concepts and whether there are factual errors or conceptual biases. This places higher demands on the model's semantic understanding capabilities and scoring consistency.

Early AAG methods primarily relied on manually generated features, rule templates, or traditional machine learning models(Heilman and Madnani, 2013; Jimenez et al., 2013; Ott et al., 2013). These methods performed well in specific domains and with fixed question types, but their generalization ability was significantly insufficient when faced with diverse expressions, cross-question types, or interdisciplinary scenarios (Saha et al., 2018). Subsequently, deep learning-based representation learning methods (such as RNNs and Transformers) significantly improved scoring performance through end-to-end training (Ahmed et al., 2022; Zhu et al., 2022).

In recent years, large language models (LLMs) have been widely used in tasks such as automatic scoring and educational feedback due to their powerful context modeling and semantic understanding capabilities. However, existing research has shown that there are still several key limitations to using LLMs directly or with simple fine-tuning for answer scoring: (1) the model is highly sensitive to language fluency and expression style, and is prone to ignoring the omission of key concepts or factual errors (Flodén, 2025); (2) the scoring results lack a consistent scale structure across different samples and different questions, resulting in insufficient stability and comparability of the scores (Jiang and Bosch, 2024; Chang and Ginter, 2024); (3) although LLMs have strong text understanding capabilities, their internal semantic representations are not specifically optimized for scoring differences, making it difficult to effectively characterize the fine-grained distinctions between different score levels (Turpin et al., 2023).

To address the aforementioned issues, this pa-

Findings of the Association for Computational Linguistics: ACL 2026, pages 19576–19589

per argues that automatic grading models need not only to understand the text, but also to explicitly perceive the degree of difference between student answers and reference answers. Based on this motivation, we propose a difference-aware contrastive learning framework for automatic answer grading with multi-level semantic modeling. Specifically, We first use heuristic semantic similarity to generate discrete difference labels between student answers and reference answers. These labels are embedded as structured prompts into the model input. This design guides the model to focus on content-level deviations rather than surface expression differences. Furthermore, we introduce a dual contrastive learning mechanism. The first component uses InfoNCE loss to enforce representation consistency among samples with the same score. The second component applies hierarchical contrastive constraints to model score gaps explicitly.

The main contributions of this paper can be summarized as follows:

We propose a difference-aware label generation prompt mechanism, which uses automatically generated difference labels to explicitly characterize the degree of semantic deviation between student answers and reference answers, enhancing the sensitivity of large-scale models to key errors and omissions.

A dual-contrast learning framework is designed to guide the model to learn a semantic representation structure that conforms to a scoring scale while maintaining classification performance.

Experiments on standard AAG datasets such as SciEntsBank and Beetle validate the effectiveness and stability of the proposed method across multiple evaluation metrics, especially demonstrating good generalization ability under small-scale model conditions.

2 Related Work

Automatic Answer Grading (AAG) has undergone rapid development, evolving from shallow models based on feature engineering and deep semantic modeling to the recent generative scoring driven by Large Language Models (LLM). Despite significant progress, existing methods still face considerable bottlenecks in multi-level semantic bias modeling, fine-grained difference perception, and score consistency. This section reviews three main research directions: traditional machine learning methods, deep learning and language modeling methods, and the emerging LLM evaluation and difference modeling methods.

2.1 Methods Based on Feature Engineering and Shallow Models

Early research primarily relied on feature engineering, using surface matching or lexical-level alignment to achieve scoring(Heilman and Madnani, 2013; Jimenez et al., 2013; Ott et al., 2013). (Mohler and Mihalcea, 2009) used semantic similarity and syntactic features to construct a regression model for automatic short answer scoring; (Saha et al., 2018) propose an approach combining token and sentence level features for short answer grading. By using the proposed features, they show that the method can overcome the limited accuracy of token level features and also the domain dependence of sentence level features.(Mohler et al., 2011) combined multiple graph alignment features and used machine learning techniques to combine semantic lexical similarity. Compared with using semantic metrics alone, this method can more accurately evaluate the student's answer; (Basu et al., 2013) improved the efficiency of manual scoring by combining clustering and rules. Furthermore, (Sultan et al., 2016) used a word alignment-based similarity scoring method to handle factual questions; (Dzikovska et al., 2013) evaluated numerous baseline models on the SemEval task, demonstrating the limitations of traditional methods in complex language expression scenarios. Overall, these methods rely on manually generated features and struggle to handle deep semantics and reasoning structures.

2.2 Methods Base on Deep Learning and Pre-trained Model

With the development of deep neural networks (Li et al., 2021), RNNs, BiLSTMs,Transformers have gained stronger sentence-level semantic modeling capabilities. (Ahmed et al., 2022) used deep networks for short answer scoring, significantly improving semantic matching ability; Siamese-LSTM (Mueller and Thyagarajan, 2016) is widely used in student answer similarity estimation; (Wang and Jiang, 2017) proposed a matching network that provides an efficient structure for complex question-answering matching.

The emergence of pre-trained language models has further promoted the progress of AAG . (Zhu et al., 2022) used BERT for short answer scoring, and the results showed that it significantly outper-

formed traditional methods on the scoring benchmark; (Aljuaid et al., 2025) used a Transformer-based graph neural network for multi-dimensional essay scoring; at the same time, a large number of studies have been conducted on BERT-based variant models. (Sayeed and Gupta, 2022; Vidula et al., 2024) However, existing BERT pre-training model methods mainly employ global vector regression, neglecting the structured differences between the answer and the Reference answer. For example, they lack explicit modeling of fine-grained error types such as missing content, logical jumps, and factual bias, which affects the interpretability and accuracy of the scores.

2.3 Contrastive Learning and Difference Perception Representation Methods

Contrastive learning, as an important method of representation learning, has been proven to have the advantage of enhancing discriminative ability in text semantic modeling. The emergence of SimCLR (Chen et al., 2020) and InfoNCE (van den Oord et al., 2019) has promoted the popularization of contrastive learning in NLP.

In answer scoring, (Zhang et al., 2022) applied contrastive learning to educational representation learning at the knowledge point level; (Mukti et al., 2023) used multi-level semantic contrastive modeling to model student answer variation patterns. However, these methods still mainly focus on "inter-sample similarity" rather than "the semantic difference structure between answer and reference answer," and cannot capture fine-grained difference features directly related to score changes.

2.4 Large Language Model Reasoning and Scoring

With the rise of LLM such as GPT-4, Claude, and Qwen, generative scoring (LLM-as-a-Grader) has become a new focus of research(Arabzadeh and Clarke, 2025). (Huang and Wilson, 2025) showed that the GPT series approaches human grader consistency in essay and short-answer question scoring; (Lee et al.) explored how CoT reasoning structures help with score interpretability. Nevertheless, numerous recent studies have shown that LLMs suffer from structural deficiencies in scoring tasks. For example, (Flodén, 2025) demonstrated that LLMs tend to avoid giving excessively high or low scores, thus failing to accurately score completely correct or completely incorrect answers; (Turpin et al., 2023) pointed out that LLMs ignore core errors in factual tasks, producing "illusory high scores." These studies indicate that while LLMs can understand text, their internal semantic space lacks specialized optimization for scoring discrepancies, thus requiring additional explicit discrepancy modeling and structured contrastive learning mechanisms.

3 Methods

To explicitly model the semantic deviation between student answer and reference answer, this study constructs a heuristic difference scoring function to generate five-level difference labels [DIFF0]-[DIFF4]. This function comprehensively considers three dimensions: word overlap rate (Jaccard Similarity), word-level edit distance (Ristad and Yianilos, 2002), and negation word difference, to achieve sensitive judgment of semantic consistency and critical errors. We use a large language model as the encoder. The hidden representations are mapped to five score levels through a linear classification head. Simultaneously, the model outputs the hidden states of all layers for use by the contrastive learning module. The framework diagram is shown in Figure 1.

3.1 Difference Marker Generation

Jaccard Similarity: The Jaccard (Niwattanakul et al., 2013) similarity between the reference answer and the student answer is defined as:

$$Jaccard(\mathrm{R},\ \mathrm{S}) = \frac{|T_R \cap T_S|}{|T_R \cup T_S|} \tag{1}$$

Where T_R and T_S denote the word sets of the reference answer and the student answer, respectively, and the Jaccard reflects the word similarity between the two answers, which is used to make a coarse-grained judgment on whether key concepts are covered in the scoring.

Edit Distance: (Ristad and Yianilos, 2002) To model deviations in semantic structure and expression of answers, we approach edit distance, calculating word sequence-based edit distances and normalizing them to [0,1] to ensure comparability of texts of different lengths:

$$Lev(\mathrm{R},\ \mathrm{S}) = \frac{Edit(T_R, T_S)}{max(|T_R| \cup |T_S|)} \tag{2}$$

Where $Edit(\cdot)$ denotes the edit distance. Detailed modeling methods are provided in Appendix A. To ensure consistency with similarity in direction, the similarity format is defined as follows:

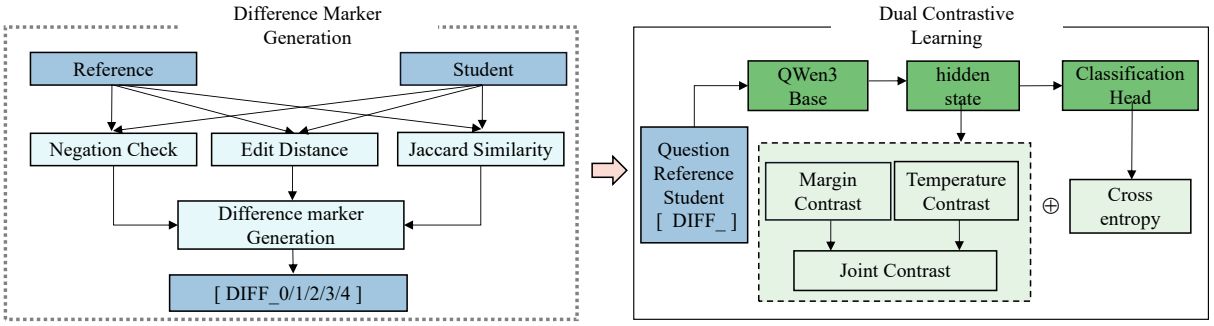


Figure 1: The overview of the proposed DiffCL framework.

$$LevSim(\mathrm{R},\ \mathrm{S}) = 1 - Lev(\mathrm{R},\ \mathrm{S}) \tag{3}$$

Negation check: Since key negative words (such as not, no, never and n′t) significantly affect the meaning of the answer, we introduced the detection of negative words to assist the large model in understanding the answer. Therefore, we define:

$$Neg(T) = \begin{cases} 1 & \text{if } T \text{ contains negative word} \\ 0 & \text{else} \end{cases} \tag{4}$$

If a student's answer contains a negative answer, but the reference answer does not, it is considered a severe semantic reversal:

$$Pen(\mathrm{R},\ \mathrm{S}) = \begin{cases} 1 & Neg(T_S) = 1 \wedge Neg(T_R) = 0 \\ 1 & Neg(T_S) = 0 \wedge Neg(T_R) = 1 \\ 0 & \text{else} \end{cases} \tag{5}$$

Jaccard measures lexical coverage, while edit distance similarity measures structural deviation:

$$Sim(R,S) = \frac{Jaccard(R,S) + LevSim(R,S)}{2} \tag{6}$$

Based on the three-dimensional word overlap rate, word-level edit distance, and negation word difference, difference labels [DIFF0]-[DIFF4] are generated:

$$Diff = \begin{cases} [DIFF0] & Sim \geq 0.80 \\ [DIFF1] & 0.60 \leq Sim < 0.80 \\ [DIFF2] & 0.35 \leq Sim < 0.60 \\ [DIFF3] & 0.2 \leq Sim < 0.35 \\ [DIFF4] & Sim \leq 0.20 \end{cases} \tag{7}$$

If a negative word penalty exists, the grade is directly downgraded to [DIFF4].

3.2 Grading Model

This study uses a large language model (such as the Qwen series) as the basic encoder and maps the hidden vectors to five-class scores (0–4) through a linear classification head. Specifically, the question, reference answer, student answer, and difference label quadruple (q, r, a, Diff) are organized as the scoring instructions input to the large language model. Simultaneously, the model outputs the hidden states of all layers for use by the contrastive learning module:

$$y_i, h_i = LLM(Q, T_R, T_S, Diff) \tag{8}$$

where y_i and h_i denote the model output and the hidden state output, respectively.

The model is trained under supervised conditions using standard cross-entropy loss:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{N} \log p(y_i|x_i) \tag{9}$$

3.3 Dual Contrastive Learning

To enhance the sensitivity of the model to the score distance, two types of contrastive learning loss are introduced.

Inspired by InfoNCE (van den Oord et al., 2019), we designed a contrast loss based on temperature parameters to encourage answers with the same score to be similar and answers with different scores to be dissimilar:

$$\mathcal{L}_{nce} = -\frac{1}{N}\sum_{i} \log \frac{\sum_{j:y_j=y_i} e^{sim(h_i,h_j)/\tau}}{\sum_{k} e^{sim(h_i,h_k)/\tau}} \tag{10}$$

Let τ denote the temperature parameter, and sim represent the cosine similarity. The index i refers to the current sample, while j denotes the other

samples sharing the same label as i (positive samples). The index k represents all samples within the batch.

To further model the score space structure, we also designed a margin loss based on score differences:

- If the difference between two sample scores is less than or equal to 1, they should be brought closer together.
- If the difference is greater than 1, they should be kept at a margin.

The sum can be expressed in terms of two points as follows:

$$\mathcal{L}_{\text{dist}} = \begin{cases} 1 - \text{sim}(h_i, h_j), & |y_i - y_j| \leq 1 \\ \max(0, \text{sim}(h_i, h_j) - 0.5), & |y_i - y_j| > 1 \end{cases} \tag{11}$$

The final training objective consists of three parts:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \left(\mathcal{L}_{nce} + \mathcal{L}_{dist} \right) \tag{12}$$

Where $\lambda \in [0, 1]$ is the weight of the contrastive loss. The value of λ was analyzed in the experimental section.

4 Experiment

4.1 Datasets

This study utilizes three representative datasets for automated answer grading, namely SciEntsbank, Beetle (Dzikovska et al., 2013). These datasets encompass a range of scoring dimensions and evaluation scenarios, providing a robust foundation for the empirical evaluation of the proposed method.

- **SciEntsbank Dataset:** This dataset is a widely recognized benchmark in short-answer automatic scoring. It contains a total of 4,969 training examples, along with 540 unseen answers, 733 unseen questions, and 4,562 unseen domains. The dataset is designed to evaluate model performance across various generalization challenges, including:
  - *Unseen Answers (UA):* Test answers that were not part of the training set;
  - *Unseen Questions (UQ):* Test questions that were not encountered during training;
  - *Unseen Domains(UD):* Test questions that originate from domains not represented in the training set.
- **Beetle Dataset:** Another widely used benchmark dataset in short-answer scoring, Beetle contains 3,941 training examples, with 439 unseen answers and 819 unseen questions. It is specifically designed to assess model robustness across unseen answer and question scenarios, but does not include the *Unseen Domains* task.

The answers in these datasets are annotated using a five-point scale: Correct, Partially Correct, Contradictory, Irrelevant, and Non-Domain.

4.2 Experimental Setup

Experiments were conducted on an NVIDIA A800 GPU. The model was trained with a learning rate of 2.5×10^{-5} for 5 epochs, chosen to ensure stable convergence. The model achieving the highest accuracy on the validation set was saved, ensuring optimal performance. The training was implemented using the Adam optimizer in the PyTorch framework.

4.3 Results

4.3.1 Overall Performance

Tables 1 and 2 present the performance comparison results of different models on the SciEntsBank and Beetle datasets, respectively. Results show that the proposed DiffCL method outperforms existing approaches on both datasets and across multiple test tasks (UA, UQ, UD), thereby validating the effectiveness and generalization capability of the proposed method.

On the SciEntsBank dataset, Our DiffCL ranks first in all evaluation tasks and metrics. Compared with traditional feature engineering methods (such as CoMeT (Ott et al., 2013), ETS (Heilman and Madnani, 2013), and SOFTCAR (Jimenez et al., 2013)), conventional machine learning models (Sultan et al., 2016; Saha et al., 2018), zero-shot on gpt-4.1-2025-04-14 [1],and Semantic Feature-Wise Transformation Relation Network (SFRN+) (Li et al., 2021), our method achieves substantial improvements in ACC, Macro-F1, and Weighted-F1. For example, on the UA task, the ACC reaches 0.7704, representing an improvement of more than

[1] https://platform.openai.com/docs/models/gpt-4.1

Model	UA			UQ			UD		
	ACC	M-F1	W-F1	ACC	M-F1	W-F1	ACC	M-F1	W-F1
CoMeT	0.6000	0.4410	0.5980	0.4370	0.1610	0.2990	0.4210	0.1210	0.2520
ETS	0.6430	0.4780	0.6400	0.4320	0.2630	0.4110	0.4110	0.3800	0.4140
SOFTCAR	0.5440	0.3800	0.5370	0.5250	0.3070	0.4920	0.5120	0.3000	0.4710
Sultan et al	0.4898	0.3298	0.4875	0.4808	0.3020	0.4676	0.5065	0.3440	0.4847
Swarnadee et al	0.6444	0.4808	0.6420	0.5007	0.3168	0.4881	0.5088	0.3574	0.4923
GPT-4 (zero-shot)	0.6340	0.4507	0.4989	0.6199	0.4801	0.5389	0.61905	0.4602	0.5290
SFRN+	0.6900	0.4700	-	0.4700	0.3500	-	0.5100	0.3500	-
Our DiffCL(Qwen3-8B)	**0.7704**	**0.6951**	**0.7621**	**0.6565**	**0.7242**	**0.6444**	**0.6981**	**0.6854**	**0.6930**

Table 1: Performance of different models on SciEntsbank dataset.

Model	UA			UQ		
	ACC	M-F1	W-F1	ACC	M-F1	W-F1
CoMeT	-	0.5690	0.6750	-	0.3000	0.4450
ETS	-	0.4440	0.5520	-	0.4610	0.5470
SOFTCAR	-	0.4550	0.5580	-	0.4360	0.4500
Sultan et al	-	-	-	-	-	-
Swarnadeep et al	-	-	-	-	-	-
GPT-4 (zero-shot)	0.7242	0.5807	0.6491	0.6399	0.5402	0.5789
SFRN+	0.7500	0.5600	-	0.6000	0.5500	-
Our DiffCL(Qwen3-4B)	**0.7904**	**0.6898**	**0.7994**	**0.6691**	**0.6728**	**0.6610**

Table 2: Performance of different models on Beetle dataset.

12 percentage points compared to the strongest baseline. On the UQ and UD task, the Weighted-F1 scores further improve to 0.6444 and 0.6930, respectively, indicating that the proposed model exhibits stronger discriminative capability and greater stability under varying scoring distributions and question types.

On the Beetle dataset, Our DiffCL also achieves better results than all compared methods, achieving the highest performance on both the UA and UQ task. Specifically, on the UA task, the ACC and Macro-F1 scores reach 0.7904 and 0.6898, respectively, demonstrating superior robustness in challenging short-answer scoring scenarios. On the UQ task, the model maintains a leading position in ACC, Macro-F1, and Weighted-F1, further confirming its advantage in cross-question generalization.

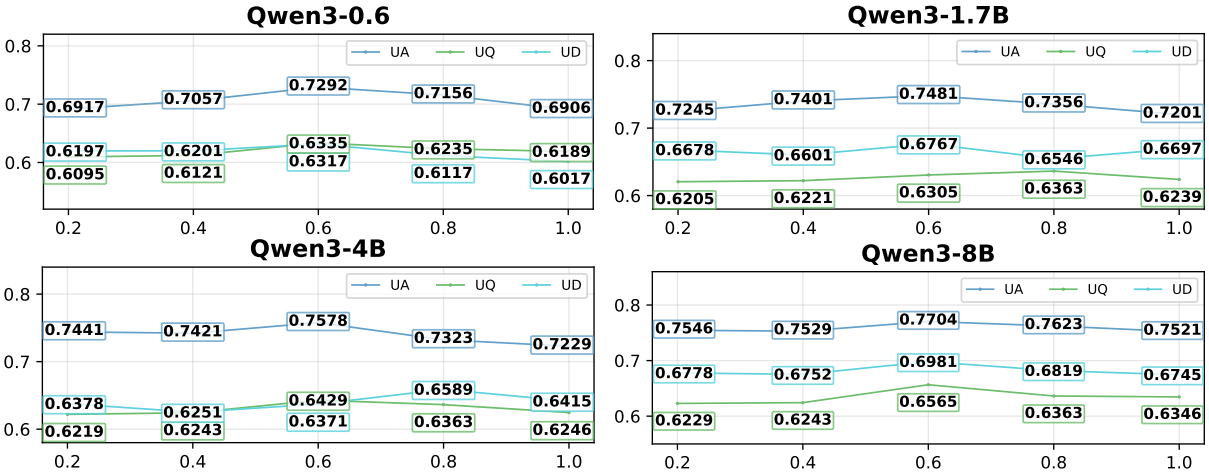
Overall, the experimental results indicate that the difference-aware mechanism (Diff Token) effectively enhances the model's ability to capture key deviations between student answers and reference answers, while the score-constrained contrastive learning strategy significantly improves the consistency and separability of scoring representations in the embedding space.

4.3.2 Contrastive Learning Ablation on Different base Models

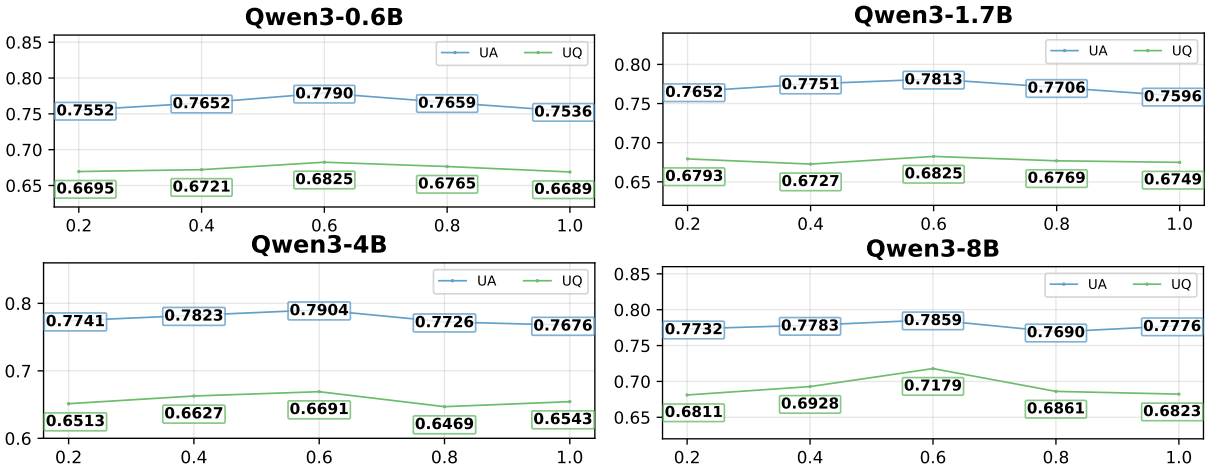
Tables 3 and 4 report the performance of the Qwen3 series models (Team, 2025) with different parameter sizes on the SciEntsBank and Beetle datasets under various loss function configurations. CE represents the standard cross-entropy loss function, TP represents the Temperature Contrast, and MG represents the Margin Contrast. Overall, augmenting the cross-entropy objective with contrastive losses consistently improves performance across evaluation tasks.

On SciEntsBank, combining all loss components (ALL) yields clear gains for both small and large models. For example, Qwen3-0.6B improves UA task accuracy from 0.7022 to 0.7292 and Macro-F1 from 0.5367 to 0.5793, while for Qwen3-8B, UA task accuracy increases from 0.7604 to 0.7704. Similar trends are observed on Beetle, where ALL improves UA accuracy from 0.7781 to 0.7904 and Macro-F1 from 0.6524 to 0.6898 on Qwen3-4B, and remains effective even for the smallest model.

Taken together, these results indicate that the temperature contrastive loss primarily strengthens intra-score consistency, whereas the margin contrastive loss encourages clearer separation between different score levels. Their combination leads to complementary effects, resulting in more structured and discriminative representations for auto-



(a) SciEntsbank dataset.



(b) Beetle dataset.

Figure 2: Model accuracy under different λ values on the SciEntsbank and Beetle dataset.

matic short-answer grading.

4.3.3 Contrast loss weighted λ sensitivity analysis

To identify the optimal weight λ for the contrastive loss, we conduct a systematic hyper-parameter study by varying λ within the range [0, 1]. Experimental results on both the SciEntsBank and Beetle datasets show a consistent trend across model sizes in the UA task: performance improves as λ increases, reaches a peak, and then gradually declines. The best overall accuracy is achieved when $\lambda = 0.6$, while larger values ($\lambda > 0.6$) lead to noticeable performance degradation, suggesting that excessively strong contrastive constraints may interfere with the primary scoring objective.

On the SciEntsBank dataset, the Qwen3-1.7B model attains its highest accuracy of 0.6589 on the UQ task at $\lambda = 0.8$, while the UA task reaches its best performance (0.6363) at the same setting. Overall, both datasets exhibit a similar "increase-then-decrease" pattern with respect to λ, indicating that a moderate contrastive loss weight is crucial for balancing representation discrimination and task-specific learning.

5 Conclusion

In this paper, we present a Difference-Aware Contrastive Learning framework for Automated Answer Grading (AAG), addressing key limitations of existing large language model–based approaches. By introducing automatically generated difference labels as explicit prompts, our method enables fine-grained perception of semantic deviations between student and reference answers. Moreover, the dual-contrastive learning mechanism, combining label-consistency InfoNCE loss and hierarchical contrastive constraints, constructs a semantically structured representation space aligned with scoring levels. Extensive experiments on benchmark datasets, including SciEntsBank and Beetle,

Model	Loss	UA			UQ			UD		
		ACC	M-F1	W-F1	**ACC**	M-F1	W-F1	**ACC**	M-F1	W-F1
Qwen3-0.6B	CE	0.7022	0.5367	0.7052	0.6138	0.6126	0.6125	0.6216	0.4549	0.6372
	CE+TP	0.7112	0.549	0.7115	0.6212	0.6230	0.6169	0.6237	0.4588	0.6372
	CE+MG	0.722	0.5591	0.7150	0.6135	0.6306	0.6201	0.6138	0.4621	0.6372
	ALL	**0.7292**	**0.5793**	**0.7252**	**0.6335**	**0.6460**	**0.6265**	**0.6317**	**0.4720**	**0.6372**
Qwen3-1.7B	CE	0.7338	0.5745	0.7392	0.6287	0.6181	0.60947	0.6660	0.5098	0.6598
	CE+TP	0.7321	0.5784	0.7308	0.6193	0.6207	0.6154	0.6709	0.5134	0.6662
	CE+MG	0.7239	0.5641	0.7310	0.6236	0.6219	0.6098	0.6688	0.5021	0.6562
	ALL	**0.7481**	**0.5844**	**0.7407**	**0.6363**	**0.6287**	**0.6246**	**0.6767**	**0.5191**	**0.6727**
Qwen3-4B	CE	0.7391	0.6849	0.7201	0.6201	0.6679	0.6193	0.6481	0.4933	0.6298
	CE+TP	0.7469	0.6899	0.7316	0.6189	0.6787	0.6194	0.6398	0.4893	0.6331
	CE+MG	0.7427	0.6914	0.7218	0.6228	0.6810	0.6244	0.6428	0.4996	0.6398
	ALL	**0.7578**	**0.7041**	**0.7486**	**0.6429**	**0.6974**	**0.6343**	**0.6589**	**0.5139**	**0.6538**
Qwen3-8B	CE	0.7604	0.6701	0.7482	0.6365	0.7014	0.6214	0.6708	0.6615	0.6710
	CE+TP	0.7567	0.6721	0.7521	0.6462	0.7112	0.6232	0.6781	0.6715	0.6789
	CE+MG	0.7510	0.6755	0.7571	0.6412	0.7156	0.6248	0.6711	0.6758	0.6791
	ALL	**0.7704**	**0.6951**	**0.7621**	**0.6565**	**0.7242**	**0.6444**	**0.6981**	**0.6854**	**0.6930**

Table 3: Performance of different base models and contrastive learning ablation on the SciEntsbank dataset

Model	Loss	UA			UQ		
		ACC	M-F1	W-F1	**ACC**	M-F1	W-F1
Qwen3-0.6B	CE	0.7554	0.6605	0.7731	0.6742	0.6820	0.6655
	CE+TP	0.7773	0.6712	0.7794	0.6657	0.6959	0.6701
	CE+MG	0.7599	0.6690	0.7744	0.6802	0.6855	0.6727
	ALL	**0.7790**	**0.6728**	**0.7883**	**0.6825**	**0.7035**	**0.6800**
Qwen3-1.7B	CE	0.7748	0.6564	0.7805	0.6689	0.6974	0.6732
	CE+TP	0.7675	0.6740	0.7885	0.6700	0.6931	0.6726
	CE+MG	0.7714	0.6601	0.7788	0.6640	0.7019	0.6615
	ALL	**0.7813**	**0.6792**	**0.7991**	**0.6825**	**0.7070**	**0.6764**
Qwen3-4B	CE	0.7781	0.6852	0.7849	0.6631	0.6555	0.6568
	CE+TP	0.7737	0.6729	0.7800	0.6563	0.6569	0.6411
	CE+MG	0.7813	0.6810	0.7900	0.6682	0.6548	0.6467
	ALL	**0.7904**	**0.6898**	**0.7994**	**0.6691**	**0.6728**	**0.6610**
Qwen3-8B	CE	0.7768	0.6710	0.7772	0.6973	0.7068	0.6948
	CE+TP	0.7808	0.6708	0.7982	0.7086	0.7128	0.7108
	CE+MG	0.7791	0.6597	0.7832	0.7001	0.7099	0.6946
	ALL	**0.7859**	**0.6772**	**0.7999**	**0.7179**	**0.7243**	**0.7125**

Table 4: Performance of different base models and contrastive learning ablation on the Beetle dataset

demonstrate that our approach consistently outperforms cross-entropy–based baselines in accuracy, weighted accuracy, and relevance metrics. Further analyses show enhanced robustness and generalization, even for small-scale models, highlighting the potential of difference-aware contrastive learning to improve reliability and interpretability in automated grading systems. Future work may explore extending this framework to cross-disciplinary and multilingual AAG scenarios, as well as integrating adaptive feedback generation to further support personalized learning.

6 Limitation

While the proposed method shows consistent improvements on benchmark AAG datasets, several limitations should be noted. First, the difference labels used in this work are derived from heuristic similarity-based rules rather than human annotations, which may not fully capture nuanced semantic deviations in complex or multi-step answers. Second, our approach focuses on modeling answer-level semantic differences and does not explicitly represent underlying concept dependencies or reasoning processes, which could be beneficial for tasks requiring fine-grained interpretability. Finally, although experiments demonstrate stable per-

formance across datasets and model scales, the evaluation is limited to a small number of English AAG benchmarks, and the generality of the method to other domains, languages, or grading rubrics remains to be validated.

7 Acknowledgments

This work was supported in part by the National Key R&D Program of China (2022YFC3303604), the National Natural Science Foundation of China (62372212), and the Key Laboratory of Smart Education of Guangdong Higher Education Institutes, Jinan University (2022LSYS003).

References

Abbirah Ahmed, Arash Joorabchi, and Martin J Hayes. 2022. On deep learning approaches to automated assessment: Strategies for short answer grading. *CSEDU (2)*, pages 85–94.

Hind Aljuaid, Areej Alhothali, Ohoud Al-Zamzami, and Hussein Assalahi. 2025. Transgat: Transformer-based graph neural networks for multi-dimensional automated essay scoring. *arXiv preprint arXiv:2509.01640*.

Negar Arabzadeh and Charles L.A. Clarke. 2025. A human-ai comparative analysis of prompt sensitivity in llm-based relevance judgment. SIGIR '25, page 2784–2788, New York, NY, USA. Association for Computing Machinery.

Sumit Basu, Chuck Jacobs, and Lucy Vanderwende. 2013. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics*, 1:391–402.

Li-Hsin Chang and Filip Ginter. 2024. Automatic short answer grading for finnish with chatgpt. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23173–23181.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Myroslava O. Dzikovska, Rodney D. Nielsen, Chris Brew, Claudia Leacock, Danilo Giampiccolo, Luisa Bentivogli, Peter Clark, Ido Dagan, and Hoa Trang Dang. 2013. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Proceedings of the 7th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2013, Atlanta, Georgia, USA, June 14-15, 2013*, pages 263–274. The Association for Computer Linguistics.

Jonas Flodén. 2025. Grading exams using large language models: A comparison between human and ai grading of exams in higher education using chatgpt. *British educational research journal*, 51(1):201–224.

Michael Heilman and Nitin Madnani. 2013. ETS: Domain adaptation and stacking for short answer scoring. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 275–279, Atlanta, Georgia, USA. Association for Computational Linguistics.

Yue Huang and Joshua Wilson. 2025. Evaluating LLM-based automated essay scoring: Accuracy, fairness, and validity. In *Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con): Works in Progress*, pages 71–83, Wyndham Grand Pittsburgh, Downtown, Pittsburgh, Pennsylvania, United States. National Council on Measurement in Education (NCME).

Lan Jiang and Nigel Bosch. 2024. Short answer scoring with gpt-4. In *Proceedings of the Eleventh ACM Conference on Learning@ Scale*, pages 438–442.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2013. SOFTCARDINALITY: Hierarchical text overlap for student response analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 280–284, Atlanta, Georgia, USA. Association for Computational Linguistics.

G G Lee, E Latif, X Wu, N Liu, and X Zhai. Applying large language models and chain-of-thought for automatic scoring. *Computers and education*.

Zhaohui Li, Yajur Tomar, and Rebecca J. Passonneau. 2021. A semantic feature-wise transformation relation network for automatic short answer grading. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6030–6040, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Michael Mohler, Razvan Bunescu, and Rada Mihalcea. 2011. 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*, pages 752–762, Portland, Oregon, USA. Association for Computational Linguistics.

Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 567–575, Athens, Greece. Association for Computational Linguistics.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Aldo Arya Saka Mukti, Syukron Abu Ishaq Alfarozi, and Sri Suning Kusumawardani. 2023. Transformers based automated short answer grading with contrastive learning for indonesian language. In *2023 15th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pages 133–138.

Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384.

Niels Ott, Ramon Ziai, Michael Hahn, and Detmar Meurers. 2013. CoMeT: Integrating different levels of linguistic modeling for meaning assessment. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 608–616, Atlanta, Georgia, USA. Association for Computational Linguistics.

Eric Sven Ristad and Peter N Yianilos. 2002. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532.

Swarnadeep Saha, Tejas I. Dhamecha, Smit Marvaniya, Renuka Sindhgatta, and Bikram Sengupta. 2018. Sentence level or token level features for automatic short answer grading?: Use both. In *Artificial Intelligence in Education*, pages 503–517, Cham. Springer International Publishing.

Mohammed Azam Sayeed and Deepa Gupta. 2022. Automate descriptive answer grading using reference based models. In *2022 OITS International Conference on Information Technology (OCIT)*, pages 262–267.

Md Arafat Sultan, Cristobal Salazar, and Tamara Sumner. 2016. Fast and easy short answer grading with high accuracy. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1070–1075, San Diego, California. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation learning with contrastive predictive coding. *Preprint*, arXiv:1807.03748.

N.A Vidula, T. Anisha Reddy, Sodum Manogna, Roshni M Balakrishnan, and Peeta Basa Pati. 2024. A machine learning based auto-grading model for question-based algebra problems using roberta. In *2024 4th International Conference on Intelligent Technologies (CONIT)*, pages 1–6.

Shuohang Wang and Jing Jiang. 2017. A compare-aggregate model for matching text sequences. In *International Conference on Learning Representations*.

Hengyuan Zhang, Dawei Li, Shiping Yang, and Yanran Li. 2022. Fine-grained contrastive learning for definition generation. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1001–1012, Online only. Association for Computational Linguistics.

Xinhua Zhu, Han Wu, and Lanfang Zhang. 2022. Automatic short-answer grading via bert-based deep neural networks. *IEEE Transactions on Learning Technologies*, 15(3):364–375.

A Word-Level Edit Distance

This appendix presents the formulation of the edit-distance-based structural difference used in our difference-aware modeling module.

A.1 Word-Level Tokenization

Given a reference answer R and a student answer S, both texts are tokenized into word-level sequences:

$$R = (r_1, r_2, \ldots, r_m), \quad S = (s_1, s_2, \ldots, s_n), \tag{13}$$

where r_i and s_j denote individual word tokens.

A.2 Edit Operations

We define three basic word-level edit operations:

- **Insertion**: inserting a word into the sequence;
- **Deletion**: removing a word from the sequence;
- **Substitution**: replacing one word with another.

Each operation is assigned a unit cost of 1.

Model	Strategy	UA			UQ			UD		
		ACC	M-F1	W-F1	**ACC**	M-F1	W-F1	**ACC**	M-F1	W-F1
Qwen3-0.6B	Jaccard	0.7095	0.5691	0.7208	0.6291	0.6406	0.6095	0.6153	0.4524	0.6266
	EDW	0.7221	0.5629	0.7221	0.6334	0.6359	0.6097	0.6301	0.4705	0.6297
	EDW+jaccrad	**0.7292**	**0.5793**	**0.7252**	**0.6335**	**0.6460**	**0.6265**	**0.6317**	**0.4720**	**0.6372**
Qwen3-1.7B	Jaccard	0.7461	0.5760	0.7316	0.6186	0.6217	0.6119	0.6763	0.5114	0.6570
	EDW	0.7389	0.5666	0.7288	0.6307	0.6158	0.6057	0.6706	0.5034	0.6622
	EDW+jaccrad	**0.7481**	**0.5844**	**0.7407**	**0.6363**	**0.6287**	**0.6246**	**0.6767**	**0.5191**	**0.6727**
Qwen3-4B	Jaccard	0.7394	0.6866	0.7452	0.6231	0.6861	0.6239	0.6458	0.5074	0.6386
	EDW	0.7522	0.7016	0.7417	0.6327	0.6786	0.6172	0.6471	0.5034	0.6400
	EDW+jaccrad	**0.7578**	**0.7041**	**0.7486**	**0.6429**	**0.6974**	**0.6343**	**0.6589**	**0.5139**	**0.6538**
Qwen3-8B	Jaccard	0.7598	0.6761	0.7542	0.**6606**	0.7147	0.6314	0.6955	0.6656	0.6783
	EDW	0.7622	0.6838	0.7551	0.6552	0.7048	0.6344	0.6952	0.6735	0.6847
	EDW+jaccrad	**0.7704**	**0.6951**	**0.7621**	0.6565	**0.7242**	**0.6444**	**0.6981**	**0.6854**	**0.6930**

Table 5: Performance of different difference marker generation strategies on the SciEntsbank dataset

Model	Strategy	UA			UQ		
		ACC	M-F1	W-F1	**ACC**	M-F1	W-F1
Qwen3-0.6B	Jaccard	0.7701	0.6601	0.7666	0.6717	0.6867	0.6765
	EDW	0.7694	0.6639	0.7756	0.6758	0.6988	0.6694
	EDW+jaccrad	**0.7790**	**0.6728**	**0.7883**	**0.6825**	**0.7035**	**0.6800**
Qwen3-1.7B	Jaccard	0.7690	0.6594	0.7799	0.6761	0.6873	0.6625
	EDW	0.7767	0.6768	0.7869	0.6646	0.6941	0.6629
	EDW+jaccrad	**0.7813**	**0.6792**	**0.7991**	**0.6825**	**0.7070**	**0.6764**
Qwen3-4B	Jaccard	0.7698	0.6774	0.7764	0.6522	0.6658	0.6479
	EDW	0.7777	0.6778	0.7769	0.6683	0.6641	0.6437
	EDW+jaccrad	**0.7904**	**0.6898**	**0.7994**	**0.6691**	**0.6728**	**0.6610**
Qwen3-8B	Jaccard	0.7752	0.6691	0.7997	0.7056	**0.7255**	0.7103
	EDW	0.7697	0.6733	0.7822	0.6957	0.7073	0.6932
	EDW+jaccrad	**0.7859**	**0.6772**	**0.7999**	**0.7179**	0.7243	**0.7125**

Table 6: Performance of different difference marker generation strategies on the Beetle dataset

A.3 Edit Distance

The edit distance between R and S is computed using dynamic programming:

$$\text{Edit}(R, S) = D(m, n), \tag{14}$$

where $D(i, j)$ denotes the minimum cost required to transform the prefix $(r_1, \ldots, r_i)$ into $(s_1, \ldots, s_j)$.

$$D(i,j) = \begin{cases} i, \text{if } j = 0, \\ j, \text{if } i = 0, \\ \min \begin{cases} D(i-1,j)+1, \\ D(i,j-1)+1, \\ D(i-1,j-1)+\mathbf{1}(r_i \neq s_j), \end{cases} \text{others}, \end{cases} \tag{15}$$

where $\mathbf{1}(\cdot)$ denotes the indicator function, which equals 1 if the condition holds and 0 otherwise.

A.4 Interpretation

The edit-distance-based structural difference quantifies the amount of lexical and structural modification required to transform a student answer into the reference answer. This metric complements lexical overlap measures and provides an interpretable estimate of structural deviation, which is subsequently used for difference-level inference in the main model.

B Difference marker Generation Ablation

This experiment aims to analyze the impact of five-level difference labels ([DIFF0]–[DIFF4]) generated based on Jaccard similarity and word-level edit distance (EDW) on the answer scoring performance of a large model. To verify the effectiveness of the difference label generation strategy, we conducted systematic ablation experiments on two standard datasets, SciEntsBank and Beetle. The results are shown in Tables 5 and 6.

Overall, the fusion strategy (EDW+Jaccard) achieved the best performance on most evaluation metrics, significantly outperforming strategies using Jaccard or EDW alone. This indicates that simultaneously modeling lexical overlap information and sequence structure differences helps to more

Model	level	UA			UQ			UD		
		ACC	M-F1	W-F1	**ACC**	M-F1	W-F1	**ACC**	M-F1	W-F1
Qwen3-8B	3	0.7507	0.6894	0.7234	0.6555	0.7181	0.6140	0.6725	0.6513	0.6752
	5	0.7620	0.6785	0.7528	0.6400	0.7117	0.6426	0.6850	0.6766	0.6889
	7	0.7490	0.6568	0.7427	0.6276	0.7035	0.6229	0.6681	0.6561	0.6866
	5(DiffCL)	**0.7704**	**0.6951**	**0.7621**	**0.6565**	**0.7242**	**0.6444**	**0.6981**	**0.6854**	**0.6930**

Table 7: Different levels of difference and threshold experiment On SciEntsbank datasets.

Model	level	UA			UQ		
		ACC	M-F1	W-F1	**ACC**	M-F1	W-F1
Qwen3-4B	3	0.7757	0.6514	0.7646	0.6300	0.6712	0.6070
	5	0.7884	0.6745	0.7916	0.6594	0.6611	0.6487
	7	0.7644	0.6764	0.7884	0.6558	0.6391	0.6313
	5(DiffCL)	**0.7904**	**0.6898**	**0.7994**	**0.6691**	**0.6728**	**0.6610**

Table 8: Different levels of difference and threshold experiment On Beetle datasets.

accurately characterize the semantic deviation of student answers from reference answers, thereby generating more discriminative difference labels.

On the SciEntsBank dataset, EDW+Jaccard maintained a stable advantage in all three subtasks. Taking the 4B model as an example, this strategy improves the accuracy and Macro-F1 score on the UA task to 0.7578 and 0.7041, respectively, demonstrating its ability to more effectively distinguish different levels of semantic bias in scenarios with complex semantic structures and high answer diversity.

On the Beetle dataset, the fusion strategy performs particularly well in the UA task, achieving the highest ACC, Macro-F1, and Weighted-F1 scores across different model sizes. For instance, on the 8B model, the UA accuracy reaches 0.7904, indicating that this method has stronger sensitivity and discriminative power for answers that are "semantically correct but expressively diverse."

Furthermore, the stable gains of the EDW+Jaccard strategy on models with different parameter sizes demonstrate that this discriminative modeling method has good robustness and scalability, is independent of specific model capacity, and can be applied as an independent module to various automatic scoring frameworks

In summary, ablation experiments have verified the effectiveness of the proposed heuristic difference scoring function in fine-grained semantic difference modeling, providing key support for improving the accuracy and reliability of automatic answer scoring for large models.

C Analysis of Difference-Level Thresholds

This appendix provides additional details regarding the design choices of the five-level difference labels used in the proposed difference-aware modeling framework, including threshold initialization, granularity selection, and robustness across datasets.

C.1 Threshold Initialization Strategy

The five difference levels (`DIFF0`–`DIFF4`) are derived from a continuous difference score computed based on lexical overlap and structural edit distance between the reference answer and the student answer. Rather than being arbitrarily chosen, the thresholds are empirically initialized according to the observed distribution of difference scores on the training data.

C.2 Rationale for Five-Level Granularity

We adopt a five-level difference scheme to match the granularity of the target scoring labels (0–4), following the principle that intermediate semantic variables should be aligned with the resolution of the final prediction task. A coarser design (e.g., three levels) fails to distinguish between subtle paraphrasing and partial correctness, while a finer-grained design (e.g., seven levels) introduces excessive noise due to the inherent uncertainty of heuristic difference estimation.

Moreover, the five-level design reflects common human grading cognition in educational assessment, where responses are typically categorized into fully correct, partially correct, and multiple

Model	Loss	UA			UQ			UD		
		ACC	M-F1	W-F1	ACC	M-F1	W-F1	ACC	M-F1	W-F1
Bert	CE	0.6844	0.5208	0.6720	0.5340	0.5556	0.5211	0.5448	0.5894	0.5299
	CE+TP	0.6887	0.5209	0.6718	0.5521	0.5499	0.5130	0.5492	0.5811	0.5302
	CE+MG	0.6901	0.5222	0.6721	0.5121	0.5505	0.5136	0.5512	0.5918	0.5462
	ALL	0.6998	0.5310	0.6812	0.5424	0.5656	0.5401	0.5641	0.5989	0.5679
RoBerta	CE	0.6902	0.5192	0.6723	0.5400	0.5606	0.5317	0.5518	0.5998	0.5391
	CE+TP	0.6912	0.5202	0.6823	0.5421	0.5701	0.5237	0.5611	0.6018	0.5461
	CE+MG	0.6987	0.5189	0.6800	0.5511	0.5726	0.5313	0.5691	0.6118	0.5492
	ALL	0.7102	0.5302	0.6922	0.5633	0.5860	0.5412	0.5798	0.6191	0.5590

Table 9: Performance of pre-trained models On SciEntsbank datasets.

Model	Loss	UA			UQ		
		ACC	M-F1	W-F1	ACC	M-F1	W-F1
BERT	CE	0.7114	0.6080	0.7103	0.5882	0.5922	0.6019
	CE+TP	0.7024	0.5938	0.7021	0.5734	0.5876	0.5986
	CE+MG	0.7098	0.5999	0.7088	0.5814	0.5987	0.5998
	ALL	0.7216	0.6181	0.7124	0.6081	0.5990	0.6201
RoBERTa	CE	0.7240	0.6190	0.7291	0.5991	0.6021	0.6109
	CE+TP	0.7110	0.6079	0.7161	0.5892	0.5982	0.6129
	CE+MG	0.7288	0.6209	0.7282	0.6001	0.6101	0.6132
	ALL	0.7330	0.6291	0.7381	0.6091	0.6129	0.6289

Table 10: Performance of pre-trained models On Beetle datasets.

degrees of incorrectness.

In summary, the thresholds serve as a form of weak supervision that initializes the difference-aware representation space, while the final semantic alignment is achieved through end-to-end training with classification and contrastive learning objectives.

C.3 Different levels of difference and threshold experiment

We conduct a systematic study on the impact of different difference-level partition strategies on both the SciEntsBank and Beetle datasets. Specifically, three configurations are considered: (1) a 3-level setting ([DIFF0]–[DIFF2]) with threshold boundaries of $[0, 0.33, 0.66, 1]$; (2) a 5-level setting ([DIFF0]–[DIFF4]) with thresholds $[0, 0.2, 0.4, 0.6, 0.8, 1]$; and (3) a 7-level setting ([DIFF0]–[DIFF6]) using thresholds $[0, 0.14, 0.28, 0.42, 0.56, 0.70, 0.84, 1]$. The corresponding results are reported in Tables 7 and 8, where **5 (DiffCL)** denotes the proposed method under the 5-level difference setting with the adopted thresholds.

As shown in Tables 7 and 8, a consistent pattern emerges across both datasets: the 5-level difference configuration yields superior overall performance compared to the 3-level and 7-level counterparts. This suggests that an overly coarse partition fails to capture meaningful semantic deviations between student and reference answer, whereas an excessively fine-grained division may introduce unnecessary noise and hinder effective representation learning. Moreover, when the number of difference levels is fixed to five, variations in the specific threshold values lead to only marginal performance changes, indicating that the proposed approach is relatively insensitive to precise threshold selection. These observations demonstrate that the effectiveness of the proposed difference-aware framework primarily stems from an appropriate granularity of difference modeling rather than reliance on finely tuned threshold values.

D Analysis of pre-trained models with fewer parameters

This experiment aims to verify the effectiveness of the proposed method on different pre-trained models. Specifically, we conducted experiments using BERT and RoBERTa as base models, and the relevant results are shown in Tables 9 and 10.

The experimental results show that the original BERT and RoBERTa significantly outperform the Qwen model on the answer scoring task, indicating that BERT and RoBERTa have certain limitations in capturing fine-grained semantic differences related to scoring. Furthermore, after introducing the Diff method for training, the performance of BERT and RoBERTa on the SciEntsBank and Beetle datasets is significantly improved. This result demonstrates that the difference-aware contrastive learning method can effectively distinguish between correct and incorrect answers, thereby improving the scoring accuracy of the model.