# Towards Human-Like Grading: A Unified LLM-Enhanced Framework for Subjective Question Evaluation

**Fanwei Zhu**[a,1], **Jiaxuan He**[b,1], **Xiaoxiao Chen**[c,*], **Zulong Chen**[b], **Quan Lu**[d] **and Chenrui Mei**[a]

[a]Hangzhou City University
[b]Alibaba Group
[c]Zhejiang Hospital
[d]Mashang Consumer Finance Co

**Abstract.** Automatic grading of subjective questions remains a significant challenge in examination assessment due to the diversity in question formats and the open-ended nature of student responses. Existing works primarily focus on a specific type of subjective question and lack the generality to support comprehensive exams that contain diverse question types. In this paper, we propose a unified Large Language Model (LLM)-enhanced auto-grading framework that provides human-like evaluation for all types of subjective questions across various domains. Our framework integrates four complementary modules to holistically evaluate student answers. In addition to a basic text matching module that provides a foundational assessment of content similarity, we leverage the powerful reasoning and generative capabilities of LLMs to: (1) compare *key knowledge points* extracted from both student and reference answers, (2) generate a *pseudo-question* from the student answer to assess its relevance to the original question, and (3) simulate human evaluation by identifying content-related and non-content strengths and weaknesses. Extensive experiments on both general-purpose and domain-specific datasets show that our framework consistently outperforms traditional and LLM-based baselines across multiple grading metrics. Moreover, the proposed system has been successfully deployed in real-world training and certification exams at a major e-commerce enterprise.

## 1 Introduction

Automatic examination assessment has been widely adopted in educational institutions and online testing platforms to reduce grading effort, accelerate feedback processes, and ensure unbiased evaluation [2]. Subjective questions, such as short-answer questions, problem-solving tasks, and case analyses, constitute a central component of comprehensive examinations. However, grading subjective questions is inherently more complex and labor-intensive than evaluating objective ones due to their diverse formats, ambiguous semantics, and open-ended nature.

Existing works on automatic subjective question evaluation primarily focus on specific question types. For example, automatic essay scoring (AES) [16] evaluate long-form essays using rubric-based criteria while automatic short answer grading (ASAG) [32] emphasizes factual alignment with reference answers. These methods lack the generality to grade exams containing diverse question types. Technically, even recent LLM-based efforts [8] often reduce grading to simplistic similarity comparison or prompted scoring, missing the question-adaptable, context-aware judgment of human graders—such as rewarding creativity for essays or precision for technical problems. Therefore, there remains a critical gap in building a unified grading framework capable of *handling the diversity and complexity of real-world subjective questions with human-like judgment*.

**Challenges.** Designing such a unified, comprehensive auto-grading system for all types of subjective questions involves multiple core challenges:

- *Redundancy and ambiguity.* Student answers often contain irrelevant or repetitive information, making it difficult to identify the most relevant points for accurate grading. Moreover, ambiguities in phrasing or interpretation further complicate the assessment of subjective question.
- *Holistic, multi-faceted assessment.* Unlike objective questions that have clear, determined answers, subjective responses require comprehensive assessment of both content aspects (*e.g.*, factual accuracy, completeness) and non-content aspects (*e.g.*, logical coherence, clarity, and structure), resembling the cognitive processes of human graders.
- *Weak answer alignment.* Students' answers can significantly deviate from the reference materials in terms of length, wording, and perspective, yet remain relevant and

* Corresponding Author. Email: 814441073@qq.com.
[1] Co-first authors with equal contribution.

insightful. Direct answer-to-answer alignment is therefore insufficient, as responses that appear quite different to the reference answers may still be relevant to the question in many cases.

**Proposal.** To address these challenges, we propose a *unified LLM-enhanced framework* for automatic grading of diverse subjective questions. Unlike direct LLM prompting approaches, our framework integrates multiple complementary modules that leverage LLMs' advanced language understanding and generation capabilities to enhance grading accuracy and robustness.

*First*, LLMs excel at distilling the most pertinent content while filtering redundant information. This capability motivates us to develop a *Key Points Matching Module* (KPM) that identifies the essential knowledge points from the student and reference answers for a *knowledge-level alignment*, improving grading accuracy despite answer redundancy and ambiguity (Challenge 1).

*Second*, trained on vast amounts of text data, LLMs possess robust capabilities in comprehending and evaluating nuanced language expressions. We leverage this through a *LLM-based General Evaluation Module* (LGE) that assesses answers across multiple dimensions (*e.g.*, semantic relevance, logical coherence, and clarity of expression) to simulate human graders' comprehensive judgment (Challenge 2).

*Third*, for the challenge of weak answer alignment, we introduce a novel reverse-matching strategy through a *Pseudo-Question Matching Module* (PQM). This module capitalizes on LLMs' generative capabilities to create pseudo-questions from students' answers, then evaluate their semantic alignment with the original question. Such *question-to-question* comparison overcomes the limitations of direct answer matching when student responses diverge in structure or phrasing yet remain semantically valid (Challenge 3).

*Additionally*, we incorporate a *Textual Similarity Matching Module* (TSM) that provides direct lexical comparisons between student and reference answers to ensure no critical textual details are overlooked during the semantic evaluation. Finally, a *Deep Fusion Layer* attentively integrates insights from all four modules, capturing cross-module dependencies and interactions for final scoring.

**Evaluation.** Due to the absence of public benchmarks for mixed-type subjective question assessment, we construct two novel datasets: one from educational examinations spanning four academic disciplines, and another from enterprise certification tests featuring domain-specific questions. To ensure comprehensive evaluation, we conduct extensive experiments on these datasets alongside a widely-used AES benchmark. Results show that our method consistently outperforms strong baselines across diverse question types, domains, and evaluation metrics.

In summary, our main contributions are as follows.

- We identify key limitations of existing approaches and propose a unified, LLM-enhanced framework that simulates human-like grading across all types of subjective questions.
- We develop four complementary modules that collectively capture both content and expression quality, combining semantic alignment and textual similarity to support comprehensive evaluation.
- We build and release two evaluation datasets covering diverse domains and question types, enabling rigorous benchmarking.
- We validate the effectiveness of our approach through extensive experiments and real-world deployment. Our code and datasets are released to support future research [7].

## 2 Related work

Existing works on automatic subjective question grading can be categorized into two types: Automatic Short Answer Grading (ASAG) which focuses on short, concise responses to questions with a pre-defined reference answer [32] and Automatic Essay Scoring (AES) that typically deals with longer, structured essays written on a given topic [19]. Although ASAG and AES differs in terms of the reliance on correct answers, the techniques in both generally evolves from traditional rule-based models to the state-of-the-art learning-based models.

**Automatic Short Answer Grading**. The primary goal of ASAG is to score the responses accurately against reference answers and its techniques emphasize on the semantic similarity with the reference answer. Traditional works mainly utilized handcrafted features for similarity evaluation. For instance, He *et al.* [9] proposed to integrate LSA and n-gram co-occurrence for improving the accuracy of automatic summary assessment. Das *et al.* [5] considered the string similarity, semantic similarity, and keyword similarity as the criterion and proposed a weighted multi-criteria-decision-making (MCDM) approach to integrate all these similarities for subjective assessment. Recently, deep learning approaches and LLMs have been widely explored for ASAG. For instance, Zhu *et al.* [33] developed a BERT-based neural network that integrates dynamic text encoding, a semantic refinement layer, and a novel triple-hot loss strategy to enhance semantic understanding and scoring precision. Yoon *et al.* [30] used LLM to identify the key phrases in student answers and compare them with those of reference answers for scoring. Schneider *et al.* [20] used GPT 3.5 to assess instructor answers, student answers, and their similarity, reporting observed issues with the LLM grading compared to human assessment.

**Automatic Essay Scoring.** AES aims to provide a holistic evaluation in multi-paragraph essays, focusing on writing quality, coherence, and content relevance. Traditional AES systems heavily depended on rule-based heuristics or feature-based similarity metrics [1, 4]. The availability of annotated corpora later prompted a shift toward learning-based AES approaches. Transformer-based models, such as R2BERT [29], multi-scale BERT [24], further advanced this area by leveraging pre-trained linguistic and commonsense

knowledge. LLMs have introduced new possibilities for AES through prompting techniques, allowing models to score essays with minimal supervision. For example, Lee *et al.* [15] used LLMs in a zero-shot setting, where no manually scored essays were provided for training. Mansour *et al.* [18] and Xiao *et al.* [25] investigated the effectiveness of LLMs for scoring in a few-shot setting, where a small set of labeled examples were included in the prompt to guide the scoring process. Song *et al.* [22] explored the use of open-source LLMs for AES and automated essay revising (AER), highlighting LLMs as efficient, cost-effective, and privacy-friendly solutions for AES and AER tasks.

**Comparison to our work.** Our work differs from existing studies in two aspects: *First*, while existing works focus on specific type of question, we propose a uniform auto-grading framework capable of handling all types of subjective questions. This include both questions with targeted or factual answers and essay-writing question with open-ended responses. *Second*, existing methods on the use of LLMs for auto-scoring are still in the early stages, showing poor alignment with human graders. Instead of using LLMs as a tool for scoring, we develop an LLM-enhanced method that fully explores the ability of LLMs in different evaluation aspects to simulate the human graders' consideration and thus enhance the accuracy of auto-grading.

## 3 Preliminaries and Problem

The application scenarios of subjective question auto-grading span across various domains such as education, corporate training, and specialized assessment. Subjective questions in comprehensive exams are designed to assess not only factual knowledge but also analytical skills, critical thinking, problem-solving, and the ability to apply knowledge. Automating the grading of such exams requires consideration of all possible types of subjective questions, including short answer questions (*e.g.*, definition, noun explanation), writing-based questions (*e.g.*, essay writing), and scenario questions (*e.g.*, case study). Based on the availability of correct answers, we categorize these question into two broad types: *factual-type questions* with fixed reference answers (*e.g.*, short answer question) and *open-ended questions* with only scoring rubrics (*e.g.*, essay-type question). In this paper, we aims to provide a unified solution for auto-grading across all types of subjective questions.

**Subjective question auto-grading.** Given a subjective question $Q$, a student answer $A$ to the question, and the reference materials $R$, which may include a reference answer, scoring points, or rubrics, the task is to construct a scoring model $\mathcal{F}$ that generates the predicted score $\hat{y}$ of $A$ based on $Q$ and $R$:

$$\mathcal{F}(A, Q, R) = \hat{y} \tag{1}$$

with the objective of minimizing the difference between $\hat{y}$ and the true score $y$ provided by the human grader.

## 4 LLM-powered Auto-grading Approach

### 4.1 Overall framework

The overall architecture of the proposed LLM-enhanced auto-grading framework is illustrated in Fig. 1. It consists of four key branches: (1) *Key Points Matching Module* that focuses on reducing redundancy and noise in the student answer by extracting key knowledge points from the student and reference answers with LLM; (2) *Pseudo-Question Generation and Matching Module* that utilizes LLM to generate pseudo-questions based on the student's answer, which are then compared with the original question to evaluate how well the student answer written in various formats aligns with the intent of the question; (3) *LLM-based General Evaluation Module* that leverages the understanding and reasoning capability of LLM to evaluate the alignment of the response with the reference answer across multiple dimensions; (4) *Textual Similarity Matching Module* that performs direct text matching between the student answer and reference answer using similarity matching technique, ensuring that no critical details are missing during feature extraction by the LLM. The output of each module is encoded and then integrated in a *Deep Fusion Layer* to produce the final score.

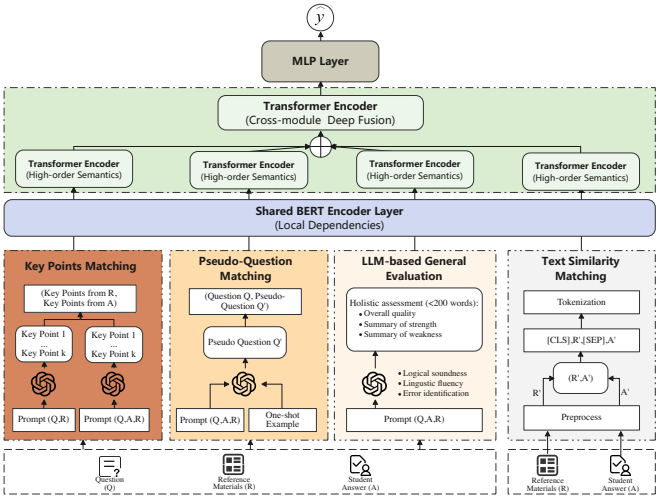


**Figure 1**: The architecture of LLM-powered auto-grading model.

### 4.2 Key Points Matching

In student answers of subjective questions, the same idea or knowledge may be expressed in diverse ways. Some may include unnecessary details, unclear phrasing, or irrelevant information that distract from the core concept. Moreover, for essay-writing type questions, reference answers are usually presented as scoring guidelines rather than standard answers (*e.g.*, 2 points for optimism, 1 point for perseverance), which makes direct comparison with reference answer even more infeasible.

To address these issues, we design a key points matching module that leverages LLM to extract the key points from

student answers and reference answers, removing redundancy and noise. The extracted key points are combined as a fused representation, which is passed into a BERT model to learn the contextual embedding for semantic matching.

*Key points extraction.* Given the question $Q$, the student answer $A$ and the reference answer $R$ as input, the LLM is prompted to analyze and extract essential knowledge points that match the intent of the question. In particular, for the student answer $A$, the LLM identifies key points that summarize the main ideas, taking into account the context provided by both the question and the reference answer. For the reference answer $R$, the LLM generates a comparable set of key points in alignment with the scoring criteria. If the reference answer only consists of scoring guidelines but no concrete answers, the guidelines are condensed and returned as key points.

This step ensures that the extracted knowledge points from both answers are concise, representative, and semantically relevant to the question. We formalize this as:

$$K_A = ExtractKeyPoints(Q, A, R) \tag{2}$$

$$K_R = ExtractKeyPoints(Q, R) \tag{3}$$

where $K_A$ and $K_R$ are the key points extracted from $A$ and $R$ respectively, typically in the form of 2-3 phrases, each within 25 characters.

*Key points pairing and encoding.* To capture the semantic correlation between student and reference answers, the extracted knowledge points are organized into a pair $< K_A, K_R >$, which is tokenized (*i.e.*, $[CLS]K_A[SEP]K_R[SEP]$) and passed into a BERT model to obtain the contextual embeddings $\mathbf{H_B}$, formalized as:

$$\mathbf{B} = BERT(< K_A, K_R >) \tag{4}$$

Here, $\mathbf{H_B} \in \mathbb{R}^{L \times d}$ where $L$ is the length of tokenized sequence and $d$ is the hidden size. While BERT captures the local semantics in the token-level embeddings, we augment it with a Transformer layer to better model high-order interactions between $K_R$ and $K_A$:

$$\mathbf{H}_K = Transformer(\mathbf{B}) \tag{5}$$

## 4.3 Pseudo-Question Generation and Matching

While knowledge points matching module focuses on the semantic similarity between student answer and reference answer to tackle the redundancy issue, we also notice that assessing answers alone cannot verify true relevance to the question in many cases. For example, the responses to open-ended questions could be highly diverse with varying length and different opinions, direct answers alignment may cause *false negatives* where student answers may appear differently to reference answer but are actually related to the question.

To reduce these risks, we introduce the Pseudo-Question Generation and Matching module, leveraging the capabilities of LLM to generate a pseudo-question based on the student's answer, capturing the intent of the response. We then assess the semantic alignment between pseudo-question and the original question, which indirectly measures whether the student answer is inherently related to the question. Such reversed matching allows a dual check of answer similarity and query relevance and improves grading accuracy.

*Pseudo-question generation with one-shot prompt.* Considering that general-purpose LLMs may face limitations when generating domain-specific questions in specialized exams, we use one-shot prompting that provides LLM with an example of question-answer pair from the same domain to guide the generation of pseudo-questions. Specifically, for a given student answer $A_i$, we first conduct a similarity search within the question pool $\mathbb{Q}$, which consists of all <question, reference answer> pairs from the dataset. We encode both student and reference answers using BERT, compute their cosine similarity, and retrieve the most relevant question-answer pair as an example, denoted as $e_i = (Q_s, A_s)$.

Next, the one-shot prompt is constructed, including the retrieved question-answer pair $e_i$ and the student answer $A_i$, to guide the LLM generate relevant and domain-aware pseduo-question $Q'_i$ for $A_i$:

$$Q'_i = LLM(e_i, A_i, Q_i) \tag{6}$$

Here, the LLM is specifically prompted to generate high-quality, semantically relevant pseudo-questions while avoiding direct retrieval of the original question $Q_i$.

*Semantic alignment of question pairs.* The generated question and original question is formed into a pair $< Q'_i, Q_i >$ for similarity alignment. This step is similar as in the key points matching module: the question pair is passed through the shared BERT layer followed by a Transformer layer to model both local and global semantic relevance between the two questions, formalized as:

$$\mathbf{H}_Q = Transformer(BERT(< Q'_i, Q_i >)) \tag{7}$$

## 4.4 LLM-based General Evaluation

The two modules described above mainly concentrates on the content-related similarity assessment, which may fails to assess non-content aspects such as overall presentation– a factor often considered by human graders, especially for open-ended question requiring structured response.

To address this, the LLM-based general evaluation module is designed to mimic human grading behavior to perform a holistic evaluation by considering not only the correctness or relevance of the student answer but also its presentation quality. In particular, given the question $Q$, the reference answer $R$ and the student answer $A$, we design prompt to guide the LLM to assess $A$ from multiple perspectives including: logical soundness, linguistic fluency and error identification. The LLM is instructed to generate a concise, holistic assessment that includes: (1) a general evaluation of the answer's overall quality (*e.g.*, very good, good, fair, poor, very poor) and (2) a summary of strengths (*e.g.*, organized presentation) and weaknesses (*e.g.*, logical flaws, inconsistencies).

The textual assessment generated by the LLM is then tokenized and passed through a BERT model and Transformer encoder to obtain a semantic presentation $\mathbf{H}_G$ for further processing.

### 4.5 Textual Similarity Matching

While LLM-based evaluation modules excel in capturing deep semantic alignment, they may overlook subtle details due to abstraction and summarization, which can be critical in human grading. Additionally, human graders typically begin by assessing textual similarity between a student's answer and the reference answer as a foundational step before delving into deeper evaluation aspects.

To address this, we integrate a direct textual matching module into our framework alongside the LLM-based modules. This serves two key purposes: (1) It simulates the initial step of human grading by providing a basic score based on textual similarity, and (2) it complements LLM-based evaluations by capturing important details that might otherwise be missed.

This module is lightweight, employing straightforward text matching techniques: First, a preprocessing step is applied to remove noise and truncate lengthy sentences in the student and reference answers. Next, the cleaned student $A'$ and reference answers $R'$ are concatenated and tokenized:

$$\mathcal{T} = Tokenizer([CLS], A', [SEP], R') \tag{8}$$

The tokenized input is then fed into a BERT encoder to generate contextual embeddings, and passed into a Transformer encoder to further model interdependencies between $A'$ and $R'$.

$$\mathbf{H}_T = Transformer(BERT(\mathcal{T})) \tag{9}$$

Example prompts for each module are provided in the appendix.

### 4.6 Cross-attention Deep Fusion and Prediction

To unify the insights of the four specialized modules into a holistic grading decision, we introduce a deep fusion and prediction layer that leverages a Transformer encoder to model cross-module dependencies and an MLP layer for final score prediction.

Specifically, the representation vectors from each module are concatenated along the sequence dimension:

$$\mathbf{H}' = Concatenate(\mathbf{H}_K, \mathbf{H}_Q, \mathbf{H}_G, \mathbf{H}_T) \tag{10}$$

where $\mathbf{H} \in \mathbb{R}^{4L \times d}$ is stacked outputs from all modules with a sequence length of $4L$.

The concatenated representation is passed into a Transformer encoder that weights and fuses the information across the modules:

$$\mathbf{H} = Transformer(\mathbf{H}') \tag{11}$$

The attention mechanism in the Transformer allow the model to capture the cross-module dependencies and attentively integrate the complementary insights from different modules.

Next, the fused embedding is compressed into a global representation through mean pooling operation and then passed into a MLP layer for the final score regression:

$$\hat{y} = Sigmoid(MLP(MeanPooling(\mathbf{H}))) \tag{12}$$

where $\hat{y}$ is the predicted score, normalized between 0 and 1 by applying the sigmoid activation.

*Loss Function.* Our model, particularly the BERT and Transformer encoders, is fine-tuned with the objective of minimizing the difference between the predicted and actual scores for grading tasks. Given a set of $N$ samples with ground-truth scores provided by human graders, we calculate the Mean Squared Error loss to quantify the difference between predicted score $\hat{y}_i$ and actual score $y_i$ as:

$$L = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{13}$$

## 5 Experiments

### 5.1 Experimental settings

**Datasets.** To comprehensively evaluate the performance of our proposed framework across different question types and domains, we conduct experiments on two *constructed datasets* covering different types of questions and a *public dataset* focusing on essay-writing questions. The characteristics of datasets is summarized in Table 1.

- **General-Type Dataset (GT)**: GT is a constructed dataset featuring a wide variety of subjective questions across four domains. It is generated using LLMs (GPT-4O for hierarchical knowledge points, Qwen-2.5-72B for question generation, and GPT-4O-Mini for answer generation), with manual validation to ensure quality. The dataset includes zero-score, full-score, and partially correct answers for each question.
- **Domain-Specific Dataset (DS)**: DS is a real-world dataset collected from an e-commerce enterprise, consisting of internal training and certification exams. The dataset includes questions from corporate compliance tests, technical skill certifications, and industry-specific assessments. It features a wide range of subjective question types, such as noun explanations, case studies and analytical questions.
- **Automated Student Assessment Prize Dataset (ASAP)** [2]: ASAP dataset is a publicly available benchmark for automated essay scoring tasks. This dataset is utilized to assess the framework's performance on essay-writing questions, particularly those lacking concrete reference answers.

[2] https://www.kaggle.com/c/asap-aes/data

All datasets are divided into training, validation and testing sets with a ratio of 10:1:1. The detailed comparison across all datasets, along with the creation process of GT dataset, can be found in Appendix A and B.

**Baselines.** To thoroughly evaluate our proposed framework, we compare it against a wide range of baselines, including two traditional similarity matching methods, nine deep learning methods, and six state-of-the-art LLM-based approaches.

- **TF-IDF + SVR** [21]: A classic regression-based model that represents both student and reference answers using TF-IDF features, followed by a Support Vector Regression (SVR) model to predict scores.
- **TF-IDF + LightGBM** [13]: Similar to the above, this method replaces SVR with LightGBM, a fast gradient boosting framework, for score prediction based on TF-IDF vectors.
- **TextCNN** [14]: A convolutional neural network trained on top of pre-trained word vectors for sentence-level classification tasks.
- **BiLSTM** [11]: A bidirectional LSTM that processes sequential data in both forward and backward directions to create richer representations and predicts scores via cosine similarity.
- **DSSM** [10]: A deep structured semantic model that encodes the reference answer and student answer independently using bert-base-chinese as the backbone. Each answer is encoded into a vector representation, and the cosine similarity between the two vectors is computed for grading.
- **BERT-base** [6]: A pre-trained bert-base-chinese model to directly assess the similarity between the student answer and the reference answer by processing them as a text pair.
- **RoBERTa** [17]: Similar to BERT, this baseline uses the pre-trained RoBERTa model to process the input text pairs and generates a similarity score between 0 and 1.
- **StructBERT** [23]: A general-purpose model for text similarity tasks. This baseline converts continuous scores into binary labels (0 or 1) and trains the model as a binary classifier to predict if the answers belong to the same class.
- **BGE-base, BGE-reranker** [26]: These models use the BGE-base-zh-v1.5 and BGE-reranker to assess the similarity between student and reference answers.
- **NPCR** [27]: A neural pairwise contrastive regression that combines regression and ranking objectives through contrastive learning to enhance scoring stability and accuracy.
- **Qwen2.5 series** [28]: Qwen2.5-72B-Instruct and Qwen2.5-14B-Instruct models are used to generate a score based on the input, which includes the question, student answer, and reference answer. In the Qwen2.5-72B-Instruct-COT variant, the model is enhanced with a Chain-of-Thought (COT) reasoning prompt, instructing the LLM to reason step-by-step before generating a score.
- **ChatGLM3-6B** [31]: A 6-billion parameter bilingual LLM developed by Tsinghua University's KEG Lab and Zhipu AI bilingual.
- **Baichuan2-13B-chat-v1** [3]: A 13-billion parameter instruction-tuned LLM developed by Baichuan Inc. for Chinese-English bilingual tasks.
- **DeepSeek-v3** [12]: A recent multilingual instruction-tuned model developed by DeepSeek AI.

**Implementation.** For data processing, we utilize the BERT Tokenizer to encode the text, with a maximum sequence length truncated to 128 tokens. During data loading, we use the DataLoader module to set a batch size of 20 for training and 64 for validation. In each LLM-based module, the Qwen2.5-72B-Instruct model is employed to generate key points, general evaluation and pseudo-questions. For model training, we use a learning rate of $2e-5$ and the AdamW optimizer. The training consists of 10 epochs, and the batch size is set to 20 for the training dataset and 64 for the validation dataset. The parameters of baselines are set as suggested in the original papers.

**Metrics.** We employ a comprehensive set of metrics to evaluate the performance of our framework on handling various grading scenarios, from regression (MSE) to binary (ACC and F1) and ordinal classification evaluations (QWK).

First, we adopt Mean Squared Error (MSE) measure the difference between the predicted score $\hat{y}$ and the ground-truth score $y$:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{14}$$

For binary scenarios (*e.g.*, pass/fail), we convert scores to binary labels ($\hat{y}_i \geq 0.5$ is correct) and report the F1 score, the harmonic mean of precision and recall, and Accuracy (ACC):

$$\text{ACC} = \frac{1}{N} = \sum_{i=1}^{N}\mathbb{I}(round(\hat{y}_i) = y_i) \tag{15}$$

where $\mathbb{I}$ is an indicator function returning 1 if the condition is true and 0 otherwise. Finally, to measure agreement with human graders in an ordinal setting, we use Quadratic Weighted Kappa (QWK) [27]. We discretize scores into five levels $[0.0, 0.25, 0.5, 0.75, 1.0]$:

$$\text{QWK} = 1 - \frac{\sum_{i,j}\mathbf{w}_{ij}\mathbf{o}_{ij}}{\sum_{i,j}\mathbf{w}_{ij}\mathbf{e}_{ij}} \tag{16}$$

where $\mathbf{o}_{ij}$ is the observe agreement (the number of cases graded as $i$ by human and $j$ by the automated system), $\mathbf{e}_{ij}$ is the expected agreement assuming random assignment, and $\mathbf{w}_{ij}$ represents quadratic weights penalizing large grading deviations. QWK ranges from -1 (worse than random) to 1 (perfect agreement), with higher value indicating better human-machine consistency.

## *5.2 Performance Analysis*

We compare our method against a wide range of baselines, and summarize the results in Table 2, where the best results are bolded and the second-best results are

**Table 1**: Summary of Datasets.

| Datasets | Question Types | Domains | Size | Language | Data Sources | Focus |
|---|---|---|---|---|---|---|
| **GT** | Mixed (short answer, essay, *etc.*) | Education Architecture Computer science Humanities | 12,000 questions | Chinese | LLM-generated with manual review | General evaluation across domains |
| **DS** | Mixed (explanation, case study, *etc.*) | Corporate compliance Technical certification | 33,724 questions | Mixed (Eng/Chn) | Enterprise training tests | Domain-specific adaptability |
| **ASAP** | student-written essays | Essays | 1,982 essays on 8 topics | English | Public benchmark dataset | Long-form essay evaluation |

underlined. Our observations and analyses are as follows: (1) *Analysis across metrics:* Our framework consistently achieves superior performance across all metrics, particularly excelling in stricter metrics MSE and QWK. For example, on the DS dataset, which features the most complex technical exam data, our model significantly outperforms the strongest learning-based baseline (NPCR) and LLM-based baseline (Qwen2.5-72B), improving the QWK by 8.5% and 31.9% respectively. (2) *Analysis across datasets:* Our model demonstrates strong and consistent performance across all datasets. On the GT dataset, where questions and answers are LLM-generated and well-structured, all models perform well- particularly LLM-based ones. Yet, our model achieves a notable MSE drop over the strongest LLM-based baseline (0.019 vs. 0.038). In the DS dataset, which poses the greatest challenge due to varied formats, mismatched answer lengths, and multi-lingual content, most models perform worse. Our model still leads, decreasing MSE to 0.080 and improving ACC, F1, QWK to 0.816, 0.843 and 0.700 respectively, highlighting its robustness in real-world assessment tasks. (3) *Analysis across LLM complexity:* Our approach maintains robust performance even with lighter LLMs (*e.g.*, Qwen2.5-7B), surpassing traditional and learning-based baselines across all datasets. This highlights the performance gains primarily stem from our module design and deep fusion strategy rather than from model scale alone.

**Latency & cost.** Our model is deployed on Alibaba Cloud with an API QPS (Questions Per Second) of 20, allowing parallel execution of all LLM modules. The total latency for grading a single question is approximately 3-5 seconds: 0.2 seconds for the BERT and Transformer layers and 3-4 seconds for the most expensive LLM module. This latency is acceptable in practical use, especially since the system is designed for asynchronous batch grading, where grading is performed in the backend and users receive the results once processing is complete.

## 5.3 Sensitivity of hyper-parameters

We analyze the impact of hyper-parameters, including the learning rate and the sequence length $L$, on the performance of our framework. Due to space limitation, we present the results only for the DS dataset, though similar trends are observed across other datasets. As shown in Fig. 2, our method performs optimally when the learning rate $r = 2e-5$ and sequence length $L = 128$. These settings are adopted for the experiments.

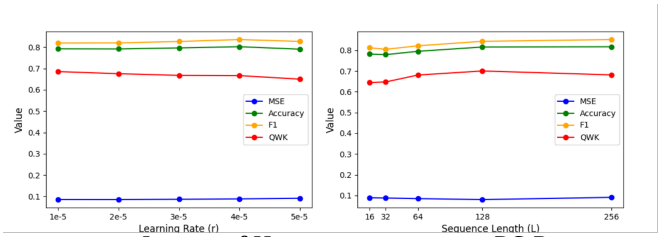


**Figure 2**: Impact of Hyper-parameters on DS Dataset.

## 5.4 Ablation Study

We conduct an ablation study to investigate the contribution of each key module. We compare our full model with five variants: (1) removing key points matching module (w/o KPM); (2) removing pseudo-question matching module (w/o PQM); (3) removing LLM general evaluation (w/o LGE); (4) removing text similarity matching module (w/o TSM); (5) replacing cross-module deep fusion with simple feature concatenation from different modules (w/o Cross).

Based on the ablation results in Table 4, we draw the following observations: (1) The full model outperforms all ablated variants, confirming that each module contributes unique perspectives on grading, and that the cross-module fusion is essential for synthesizing them into an accurate, human-like score. (2) Removing the cross-module fusion leads to the most significant performance drop across all metrics, indicating that transformer-based fusion is crucial for effectively integrating signals from all modules and producing coherent grading signals. (3) Excluding either LGE or KPM module leads to consistent performance drops. This highlights their importance in modeling semantic correctness and grading criteria at a conceptual level—key aspects of human-like evaluation. (4) The PQM and TSM modules show minor impact when removed. This is understandable in the GT dataset, where both questions and answers are LLM-generated and thus typically well-aligned. In such cases, reverse matching and text-level similarity offer limited additional gain. Nonetheless, their presence still contributes to the overall performance.

## 5.5 Case study

We further conduct a qualitative analysis through representative examples to demonstrate how our model effectively simulates human grading. Table 3 shows the three examples representing high, medium and low scoring scenarios from GT dataset while more examples on AES can be found in Appendix C.

In Case 1 (High-Scoring), the student's response effectively captured the core distinction between PPP and BOT models. Our model precisely identified these valid

**Table 2**: Comparison of Model Performance on GT, DS, and ASAP Datasets

| Category | Model | GT Dataset | | | | DS Dataset | | | | ASAP Dataset | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MSE↓ | ACC↑ | F1↑ | QWK↑ | MSE↓ | ACC↑ | F1↑ | QWK↑ | MSE↓ | ACC↑ | F1↑ | QWK↑ |
| Traditional | TF-IDF + SVR | 0.102 | 0.750 | 0.761 | 0.562 | 0.095 | 0.785 | 0.821 | 0.613 | 0.069 | 0.766 | 0.748 | 0.654 |
| | TF-IDF + LightGBM | 0.088 | 0.756 | 0.791 | 0.647 | 0.097 | 0.784 | 0.824 | 0.608 | 0.066 | 0.759 | 0.742 | 0.695 |
| Learning-based | TextCNN | 0.088 | 0.764 | 0.787 | 0.671 | 0.100 | 0.775 | 0.813 | 0.615 | 0.074 | 0.744 | 0.722 | 0.661 |
| | BiLSTM | 0.087 | 0.803 | 0.820 | 0.700 | 0.144 | 0.689 | 0.785 | 0.329 | 0.087 | 0.693 | 0.651 | 0.596 |
| | BERT-Base | 0.032 | 0.878 | 0.892 | 0.752 | 0.090 | 0.797 | 0.827 | 0.583 | 0.060 | 0.762 | 0.738 | 0.533 |
| | RoBERTa | 0.038 | 0.873 | 0.893 | 0.738 | 0.087 | 0.802 | 0.833 | 0.591 | 0.066 | 0.763 | 0.739 | 0.535 |
| | StructBERT | 0.126 | 0.776 | 0.721 | 0.613 | 0.192 | 0.646 | 0.561 | 0.328 | 0.189 | 0.755 | 0.308 | 0.196 |
| | BGE-base | 0.197 | 0.612 | 0.739 | 0.285 | 0.176 | 0.640 | 0.746 | 0.238 | 0.174 | 0.648 | 0.741 | 0.268 |
| | BGE-reranker | 0.286 | 0.672 | 0.769 | 0.267 | 0.219 | 0.694 | 0.759 | 0.372 | 0.244 | 0.681 | 0.741 | 0.374 |
| | DSSM | 0.073 | 0.805 | 0.836 | 0.728 | 0.104 | 0.767 | 0.809 | 0.611 | 0.062 | 0.780 | 0.770 | 0.734 |
| | NPCR | 0.055 | 0.823 | 0.842 | 0.667 | 0.095 | 0.774 | 0.805 | 0.615 | 0.060 | 0.792 | 0.782 | 0.749 |
| LLM-based | Qwen2.5-14B | 0.040 | 0.914 | 0.925 | 0.876 | 0.156 | 0.684 | 0.762 | 0.364 | 0.103 | 0.799 | 0.819 | 0.588 |
| | Qwen2.5-72B | 0.038 | **0.917** | 0.927 | 0.881 | 0.156 | 0.685 | 0.759 | 0.381 | 0.102 | 0.792 | 0.817 | 0.588 |
| | Qwen2.5-72B-COT | 0.041 | **0.917** | 0.927 | 0.866 | 0.156 | 0.681 | 0.759 | 0.361 | 0.096 | 0.799 | 0.817 | 0.615 |
| | Chatglm3-6B | 0.040 | 0.908 | 0.920 | 0.872 | 0.157 | 0.680 | 0.759 | 0.362 | 0.101 | 0.801 | 0.821 | 0.593 |
| | Baichuan2-13B-chat-v1 | 0.040 | 0.910 | 0.922 | 0.875 | 0.158 | 0.680 | 0.758 | 0.359 | 0.101 | 0.801 | 0.821 | 0.593 |
| | Deepseek-v3 | 0.040 | 0.907 | 0.919 | 0.872 | 0.156 | 0.682 | 0.760 | 0.362 | 0.104 | 0.796 | 0.816 | 0.581 |
| LLM-enhanced | Ours-Qwen2.5-7B | 0.024 | 0.903 | 0.913 | 0.912 | 0.088 | 0.809 | **0.845** | 0.676 | 0.060 | 0.816 | 0.816 | 0.750 |
| | Ours-Qwen2.5-14B | 0.021 | 0.898 | 0.907 | 0.915 | 0.086 | 0.802 | 0.832 | 0.681 | 0.060 | 0.817 | 0.816 | 0.760 |
| | Ours-Qwen2.5-72B | **0.019** | **0.917** | **0.928** | **0.929** | **0.080** | **0.816** | 0.843 | **0.700** | **0.059** | **0.836** | **0.840** | **0.772** |

**Table 3**: Automated Grading Examples

| | Case 1: High-Scoring Example | Case 2: Medium-Scoring Example | Case 3: Low-Scoring Example |
|---|---|---|---|
| **Question** | Explain the main differences between PPP and BOT models. | Describe the basic components and functions of a LAN. | Explain the advantages and applications of PPP in construction projects. |
| **Student Answer** | PPP emphasizes long-term collaboration with government while BOT focuses on phased transfer before transferring it to the government. | LAN consists of devices/adapters for data exchange. Its main functions are data exchange and resource sharing. | PPP improves efficiency in commercial sectors. |
| **Reference Answer** | PPP: long-term partnership; BOT: specific-phase operation. | LAN includes servers, workstations, and protocol; enables resource sharing, communication, and service sharing. | PPP advantages: risk-sharing, higher service quality, diversified funding (reduces government burden). |
| **Key Points (Student)** | PPP: long-term; BOT: transfer | Devices, adapters; data exchange | Efficiency |
| **Key Points (Reference)** | PPP: partnership; BOT: phased operation | Servers, workstations; communication; sharing | Risk-sharing; service quality; funding |
| **Pseudo-Question** | Explain differences between PPP and BOT, focusing on cooperation methods and implementation. | List LAN components and describe their primary functions. | Describe PPP's main roles and application fields in projects. |
| **LLM Evaluation** | Good: Correctly identifies core differences but lacks depth on PPP collaboration scope. | Average: Captures basic elements but misses some components like servers/protocols and functionalities. | Poor: Misses critical points (risk-sharing, funding) and misrepresents applications. |
| **Predicted / True Scores** | 0.99 / 1.0 | 0.52 / 0.5 | 0.01 / 0.0 |

**Table 4**: Results for ablation study.

| Model | **MSE ↓** | **ACC ↑** | **F1 ↑** | **QWK ↑** |
|---|---|---|---|---|
| w/o LGE | 0.022 | 0.911 | 0.920 | 0.920 |
| w/o PQM | 0.019 | 0.916 | 0.927 | 0.926 |
| w/o KPM | 0.020 | 0.907 | 0.916 | 0.923 |
| w/o TSM | 0.019 | 0.916 | 0.926 | 0.926 |
| w/o Cross | 0.020 | 0.894 | 0.905 | 0.915 |
| Full | **0.018** | **0.917** | **0.928** | **0.929** |

key points, assigning a high predicted score (0.99), closely aligning with human grading (1.0). In the medium-scoring case, the student partially covered basic LAN components (devices/adapters) and functions (data exchange, resource sharing) but omitted critical aspects like servers and communication protocols. Our model accurately identified these omissions, assigning a moderate predicted score (0.52), well-matched to the human score (0.5). In Case 3 (Low-scoring), the response provided a superficial answer about PPP efficiency without addressing crucial points like risk-sharing and service quality. The model identified these severe inadequacies, assigning a very low score (0.01) consistent with human grading (0.0).

Overall, our auto-grading approach demonstrates consistent and accurate performance in identifying critical elements across varied response quality, closely matching human evaluation standards.

### *5.6 Online A/B Test*

After a comprehensive offline evaluation, the proposed auto-grading framework has been successfully deployed on an enterprise online testing system to grade real-world tests. To assess its performance in practical scenarios, we compare it against a similarity matching-based grading baseline that utilizes BERT model for scoring.

We conduct experiments on two distinct categories of tests:

- **Value-oriented tests (D1)**: This dataset includes 100 subjective questions focused on personal reflection and value-oriented assessments. These questions are designed to evaluate candidates' compliance knowledge, decision-making skills, and alignment with personal and organizational values.
- **Technique-oriented tests (D2)**: This dataset consists of 2,000 questions aimed at assessing knowledge, problem-solving, and technical proficiency in areas such as operations, customer service, finance, and other enterprise-related tasks. The questions cover a variety of practical and scenario-based challenges, requiring multi-step reasoning and decision-making.

**Table 5**: Online Evaluation of Model Performance

| **Model** | **D1 Dataset** | | | | **D2 Dataset** | | | |
|---|---|---|---|---|---|---|---|---|
| | MSE↓ | ACC↑ | F1 ↑ | QWK↑ | MSE↓ | ACC↑ | F1 ↑ | QWK↑ |
| Baseline | 0.130 | 0.760 | 0.807 | 0.597 | 0.079 | 0.794 | 0.829 | 0.691 |
| Ours-Qwen2.5-72B | **0.125** | **0.780** | **0.823** | **0.610** | **0.078** | **0.814** | **0.844** | **0.704** |

From the results illustrated in Table 5, we can conclude that our method is consistently superior on all type of grading tasks in real online testing scenarios.

## 6 Conclusion

In this paper, we present a unified LLM-enhanced auto-grading framework that addresses the challenges of grading diverse types of subjective questions. By integrating four complementary modules, our framework provides a holistic, human-like evaluation of student answers from multiple dimensions. Experimental results on general and domain-specific datasets demonstrate that our method significantly outperforms baseline approaches across various question types. The successful deployment of our solution in real-world online testing platform within a leading e-commerce enterprise highlights its practical effectiveness and robustness.

## Acknowledgments

This work is partially supported by the Zhejiang Provincial Natural Science Foundation (No. LY24F020013) and Alibaba Innovative Research Program. The authors would like to acknowledge the Supercomputing Center of Hangzhou City University and Zhejiang Provincial Engineering Research Center for Real-Time SmartTech in Urban Security Governance, for their support of the advanced computing resources.

## References

[1] Y. Attali and J. Burstein. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3), 2006.

[2] M. M. Babitha, C. Sushma, V. K. Gudivada, et al. Trends of artificial intelligence for online exams in education. *International journal of Early Childhood special Education*, 14(01): 2457–2463, 2022.

[3] Baichuan-Inc. Baichuan 2: Open large-scale language models, 2023. https://baichuan-ai.github.io/baichuan2/.

[4] L. Bin, L. Jun, Y. Jian-Min, and Z. Qiao-Ming. Automated essay scoring using the knn algorithm. In *2008 International Conference on Computer Science and Software Engineering*, volume 1, pages 735–738. IEEE, 2008.

[5] B. Das, M. Majumder, A. A. Sekh, and S. Phadikar. Automatic question generation and answer assessment for subjective examination. *Cognitive systems research*, 72:14–22, 2022.

[6] J. Devlin. Bert: Pre- training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Z. Fanwei, H. Jiaxuan, C. Xiaoxiao, C. Zulong, l. Quan, and M. Chenrui. Code and data for "towards human-like grading: A unified llm-enhanced framework for subjective question evaluation", 2025. https://anonymous.4open.science/r/LASQ-5D80/README.md.

[8] J. Gu, X. Jiang, Z. Shi, H. Tan, X. Zhai, C. Xu, W. Li, Y. Shen, S. Ma, H. Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

[9] Y. He, S. C. Hui, and T. T. Quan. Automatic summary assessment for intelligent tutoring systems. *Computers & Education*, 53(3):890–899, 2009.

[10] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338, 2013.

[11] Z. Huang, W. Xu, and K. Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[12] D. Inc. Deepseek llm technical report. *Technical Report*, 2024. https://huggingface.co/deepseek-ai.

[13] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[14] Y. Kim. Convolutional neural networks for sentence classification. In A. Moschitti, B. Pang, and W. Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL https://aclanthology.org/D14-1181/.

[15] S. Lee, Y. Cai, D. Meng, Z. Wang, and Y. Wu. Prompting large language models for zero-shot essay scoring via multi-trait specialization. *arXiv preprint arXiv:2404.04941*, 2024.

[16] S. Li and V. Ng. Automated essay scoring: A reflection on the state of the art. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17876–17888, 2024.

[17] Y. Liu. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.

[18] W. Mansour, S. Albatarni, S. Eltanbouly, and T. Elsayed. Can large language models automatically score proficiency of written essays? *arXiv preprint arXiv:2403.06149*, 2024.

[19] D. Ramesh and S. K. Sanampudi. An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*, 55(3):2495–2527, 2022.

[20] J. Schneider, B. Schenk, and C. Niklaus. Towards llm-based autograding for short textual answers. *arXiv preprint arXiv:2309.11508*, 2023.

[21] A. Sethi. Support vector regression tutorial for machine learning. *s interneta, https://www. analyticsvidhya. com/blog/2020/03/support-vector-regression-tutorial-formachine-learning*, 3, 2020.

[22] Y. Song, Q. Zhu, H. Wang, and Q. Zheng. Automated essay scoring and revising based on open-source large language models. *IEEE Transactions on Learning Technologies*, 2024.

[23] W. Wang, B. Bi, M. Yan, C. Wu, Z. Bao, J. Xia, L. Peng, and L. Si. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv preprint arXiv:1908.04577*, 2019.

[24] Y. Wang, C. Wang, R. Li, and H. Lin. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. *arXiv preprint arXiv:2205.03835*, 2022.

[25] C. Xiao, W. Ma, S. X. Xu, K. Zhang, Y. Wang, and Q. Fu. From automation to augmentation: Large language models elevating essay scoring landscape. *arXiv preprint arXiv:2401.06431*, 2024.

[26] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie. C-pack: Packed resources for general chinese embeddings. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–649, 2024.

[27] J. Xie, K. Cai, L. Kong, J. Zhou, and W. Qu. Automated essay scoring via pairwise contrastive regression. In *Proceedings of the 29th international conference on computational linguistics*, pages 2724–2733, 2022.

[28] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[29] R. Yang, J. Cao, Z. Wen, Y. Wu, and X. He. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, 2020.

[30] S.-Y. Yoon. Short answer grading using one-shot prompting and text similarity scoring model. *arXiv preprint arXiv:2305.18638*, 2023.

[31] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, Y. Wang, et al. Glm: General language model pretraining with autoregressive blank infilling. *arXiv preprint arXiv:2210.02414*, 2022.

[32] L. Zhang, Y. Huang, X. Yang, S. Yu, and F. Zhuang. An automatic short-answer grading model for semi-open-ended questions. *Interactive learning environments*, 30(1):177–190, 2022.

[33] X. Zhu, H. Wu, and L. Zhang. Automatic short-answer grading via bert-based deep neural networks. *IEEE Transactions on Learning Technologies*, 15(3):364–375, 2022.

## A Construction of Dataset GT

To enable a comprehensive evaluation of various types of questions, we leverage LLMs to construct a dataset GT that covers four various domains: basic education (early grade Chinese, Math, and English), architecture (first-/second-class architect exams), computer science (Level 1/2 computing exams), and humanities (public welfare and arts). Each question is paired with a reference answer, which can either be a complete standard answer or a list of key scoring points, as well as multiple types of student answers, including fully correct, partially correct, and incorrect responses. The details of the dataset are summarized in Table 6.

**Table 6**: Summary of General-Type (GT) Dataset

| **Property** | **Details** |
|---|---|
| Total Questions | 12,000 questions:<br>- Train: 10,000<br>- Validation: 1,000<br>- Test: 1,000 |
| Domains | Basic Education<br>Architecture<br>Computer Science<br>Humanities |
| Reference Answer Types | Standard Answers<br>Key Scoring Points |
| Student Answer Types | Correct<br>Partially Correct<br>Incorrect |
| Scoring Scale | Normalized to [0,10] |
| Question Generation Models | Qwen-2.5-72B<br>GPT-4o-mini |
| Evaluation Models | GPT-4<br>Qwen-max |

The dataset construction process involves multiple steps:

**(1) Knowledge point generation using GPT-4o:** GPT-4o is used to generate hierarchical knowledge points for each domain. These points are then refined and expanded to create a more comprehensive set of topics.

**(2) Question generation using Qwen-2.5-72B**: Based on these knowledge points, questions are generated using Qwen-2.5-72B by setting prompts specific to each domain and topic. These prompts ensure that the generated questions are relevant, accurate, and align with the specified domain.

**(3) Answer Generation using GPT-4o-mini**: For each question, we use GPT-4o-mini to generate three types of answers: a full-score answer, a random-score answer, and a zero-score answer. The answers are designed to reflect different levels of correctness, making the dataset suitable for training, validation, and testing purposes.

**(4) Validation and Manual Review**: The generated questions and answers are further validated using stronger mod-

els such as GPT-4 and Qwen-max, followed by manual review on randomly selected samples. This dual-stage validation ensures that the content meets quality standards. Low-quality samples are filtered out to maintain the reliability and effectiveness of the dataset for training auto-grading models.

## B Comparison across Datasets

A detailed comparison of average character lengths across datasets is shown in Fig. 3. As illustrated, the three datasets present distinct characteristics that influence the complexity of the grading task. Specifically, in the*DS dataset*, student answers are notably longer than reference answers, reflecting detailed, case-based responses common in enterprise exams, while reference answers tend to be concise scoring guidelines. The *GT dataset* shows uniformly short lengths across all fields due to its LLM-generated nature and simpler question styles. In the *ASAP dataset*, reference answers are significantly longer (around 200 characters) than student essays as they include detailed rubrics and comprehensive solution description. This comparison provides important context for interpreting model performance across datasets.

## C Case studies on the ASAP dataset

We also conduct case studies on ASAP benchmark to analyze the effectiveness of our model in automatic essay scoring (AES). Unlike the GT and DS datasets, the ASAP data involves longer, free-form essays, with varied writing styles and vocabulary, offering a rigorous benchmark for human-like grading.

As illustrated in Table 7, in the high-score example, the student accurately captured crucial experimental conclusions and effectively suggested specific improvements (e.g., initial length specification and consistent weight addition). Our model precisely identified these valid key points, assigning a high predicted score (0.971), closely aligning with human grading (1.0). However, the automated evaluation also reasonably highlighted minor shortcomings, such as the student's lack of deeper reasoning about ductility differences, showcasing a nuanced and human-like grading capability.

In the medium-score example, the student's response partially addressed the similarities (specialized diets) between pandas and koalas but lacked detailed insights into their ecological specialist characteristics and the generalist traits of pythons. Our model identified the correctness of dietary habits but appropriately penalized the absence of broader ecological implications, reflected in the moderate predicted score (0.523), accurately matching the human-assigned score (0.5).

In the low-score example, the student's response was notably superficial, merely associating the term "invasive" with pythons without addressing its ecological and thematic significance (e.g., biodiversity threats and ecological balance). Consequently, our model effectively recognized this inadequacy, assigning a low predicted score (0.016), consistent with human evaluation (0.0). The automated evaluation clearly articulated these critical gaps, demonstrating a precise capability for detailed negative feedback.

Overall, the analysis of these cases demonstrates our model's robust human-like evaluation capability in AES.

## D Examples of Prompts

### *D.1 Prompts for Key Points Extraction*

Our KPM module employs carefully designed LLM prompts to extract and align key knowledge points between student responses and reference answers. This enables precise knowledge-level assessment by identifying and matching core concepts. Example prompts are illustrated in Figures 4 and 5.

### *D.2 Prompt for Pseudo-Question Generation*

In the Pseudo-Question Generation and Matching module (PQM), we design LLM prompt to generate pseudo-questions from student answers, enabling a novel reverse question-to-question matching approach. This strategy effectively addresses the limitations of direct answer-to-answer comparison. The specific prompt structure for pseudo-question generation is demonstrated in Figure 6.

### *D.3 Prompt for LLM-based General Evaluation*

We design structured prompts to guide the LLM in evaluating student answers across multiple dimensions—including both content-related (e.g., factual accuracy, completeness) and non-content aspects (e.g., logical coherence, clarity)—and generate a comprehensive yet concise assessment. The exact prompt formulation is provided in Figure 7.

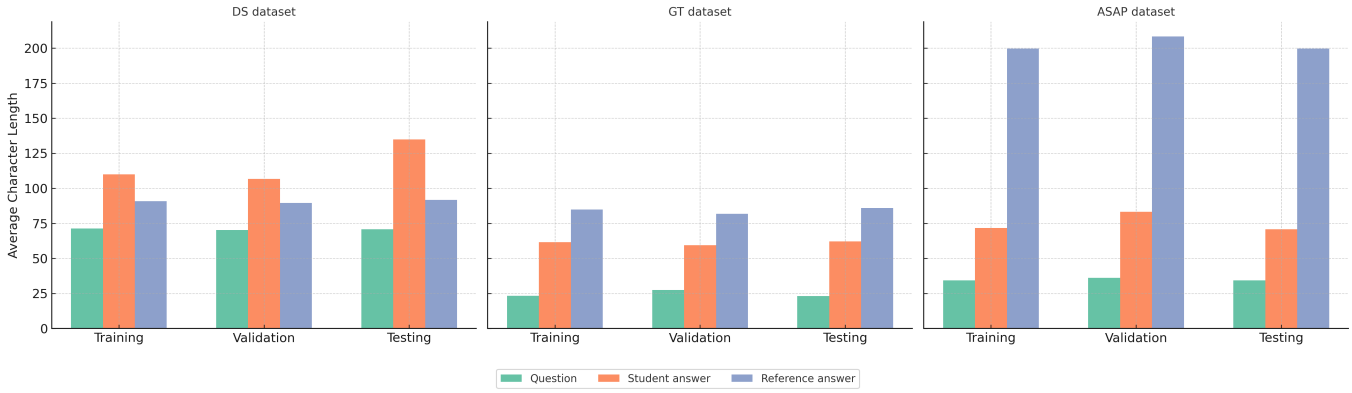


**Figure 3**: Average Character Lengths Across Datasets.

**Table 7**: Automated Grading Examples

| | **Case 1: High-Scoring Example** | **Case 2: Medium-Scoring Example** | **Case 3: Low-Scoring Example** |
|---|---|---|---|
| **Question** | Draw conclusions from students' data on four polymer plastics' ductility and suggest two experiment improvements. | Explain similarities between pandas and koalas, their differences from pythons, with supporting text. | Explain the meaning of the term "alien species" and its impact on the main theme. |
| **Student Answer** | Plastic type B stretched most, type A stretched least. Students can specify initial lengths and added weight. | Pandas and koalas are similar as they mainly eat bamboo or similar plants; snakes eat diverse animals. | "Invasiveness" describes the python, indicating its importance as an invasive species. |
| **Reference Answer** | Plastic B highest extensibility; Control thickness/initial length; More trials. | Specialists (pandas/koalas) vs generalist (python); Habitat impacts. | 'Alien' species debate: biodiversity threats vs human bias. |
| **Key Points (Student)** | Type B most ductile, type A least ductile; initial length; | Panda & koalas consume plants; snakes eat varied animals | No valid points |
| **Key Points (Reference)** | Type B highest ductility; consistent initial length; uniform thickness | Pandas and koalas are specialists; snakes are generalists; specialists adapt to specific habitats | Alien species definition; biodiversity impact; ecological issues triggered |
| **Pseudo-Question** | Draw conclusions from data on four polymer plastics' ductility and suggest two improvements. | Explain similarities between pandas and koalas, differences with pythons, using textual support. | Explain the usage of the word "invasiveness" and its impact on the main theme. |
| **LLM Evaluation** | Good: Correctly identified plastic ductility ranking and reasonable experiment improvements, but lacked discussion on applications and additional trials. | Average: Correct dietary similarity; missed specialist vs generalist elaboration and deeper implications. | Poor: Recognized basic meaning, failed to elaborate on broader ecological and thematic implications. |
| **Predicted Score** | 0.971 | 0.523 | 0.016 |
| **Ground-truth Score** | 1.0 | 0.5 | 0.0 |

LLM Prompt for Extracting knowledge Points (Student)

You are an intelligent assistant tasked with helping the examiner identify scoring points in a candidate's answer. Please follow the instructions below:
1. The candidate's response may contain irrelevant or incorrect content. Focus only on the valid scoring points.
2. Based on the exam question, reference answer, and grading criteria, extract key knowledge points from the candidate's answer.
3. If the answer contains no valid scoring points, output No scoring points.
4. Ignore redundant content. Do not provide subjective evaluations. You may condense the information slightly but must preserve the original meaning.
5. Extract 2–3 knowledge points at most. If fewer are available, extract what is present.
6. Each knowledge point should be concise—preferably no more than 25 Chinese characters.
**Instructions:**

- You will be provided with the exam question, grading criteria, and candidate's answer.
- Extract the scoring points from the candidate's answer and list them in the order they appear.
- Use concise Chinese. If there are multiple points, separate them with semicolons.

**Output Format:**

- Only output the final extracted knowledge points, with no additional comments or formatting.
- Example: `{"point1;point2;point3"}`
- If no valid scoring points are found, output: `"No valid points"`

**Exam question:** {question}
**Reference answer:** {ref answer}
**Student answer:** {std answer}

**Figure 4**: Prompt for Extracting Knowledge Points from Student Answer

LLM Prompt for Extracting knowledge Points (Reference)

You are an intelligent exam assistant tasked with helping the examiner extract core knowledge points from a reference answer. Follow the instructions below:
1. The reference answer may contain redundant or overly detailed descriptions. Focus on the essential knowledge only.
2. Based on the exam question and grading criteria, extract concise and standardized key knowledge points that directly address the exam requirements.
3. Each knowledge point should be within 25 Chinese characters and reflect core content only. Avoid redundancy or vague expressions.
4. A maximum of 2–3 knowledge points should be extracted. If there are insufficient points, fewer may be returned.
**Procedure and Output Requirements:**

- Provide the exam question and reference answer.
- List the extracted knowledge points in Chinese, in the order they appear in the reference answer.
- Separate each knowledge point using a semicolon.
- If the answer is already short and concise, return it as the only knowledge point.

**Exam question:** {question}
**Reference answer:** {ref answer}

**Figure 5**: Prompt for Extracting Key Knowledge Points from Reference Answer

LLM Pseudo-Question Generation Prompt

You are an intelligent exam assistant. Your task is to generate a pseudo-question based on a candidate's answer and compare it with the original exam question to assess the relevance and accuracy of the candidate's response.

**Pseudo-question generation requirements:**

- Identify the core issue or description presented in the candidate's answer.
- Ignore redundant information or excessive focus on incorrect points.
- The generated pseudo-question should align semantically with the main intent of the candidate's answer.

**Output format:**

- Provide only the generated pseudo-question with no extra commentary or explanation.
- Format the output as: `{"pseudo-question"}`

**Exam question:** {question}
**Reference answer:** {ref answer}
**Candidate answer:** {std answer}

**Figure 6**: Prompt for Pseudo-Question Generation Based on Student Answer

LLM Evaluation Prompt

You are an intelligent assistant responsible for evaluating a candidate's answer. Below are your job requirements:
1. Based on the exam question and reference answer, analyze the overall performance of the candidate's answer and provide a clear evaluation (e.g., "good," "average," "poor").
2. In the evaluation, point out the strengths (correct points) and weaknesses (incorrect points) of the candidate's answer to help the examiner understand the key issues in the answer.

**Specific steps and output format requirements:**

- Provide the exam question, grading criteria, reference answer, and candidate's answer.
- Extract the correct and incorrect points from the candidate's answer based on the grading criteria, and provide an overall evaluation based on these points.
- Consider aspects such as logical consistency, language fluency, and content rationality when evaluating the answer.
- **Output format:** A short description, preferably no more than 200 Chinese characters. Do not use bullet points.

**Example output:**

```
 {"The overall evaluation of the candidate's answer is good. Strengths: The candidate
listed specific instances of SMS channel violations in detail, such as misleading red
packet information and improper use of brand terms, which align with the violation
facts mentioned in the grading criteria. Additionally, the candidate mentioned that
penalties would be imposed based on the violation facts and recommended self-inspection
by the candidate, demonstrating the platform's rigor and fairness in handling issues.
Weaknesses: The candidate's answer did not mention the caution and multiple review
processes involved in penalties, and the explanation for not providing personal
evidence was insufficient, which may affect the candidate's understanding of the
penalty's reasonableness."}
```

**Exam question:** {question}
**Grading criteria:** {ref answer}
**Candidate's answer:** {std answer}

**Figure 7**: Prompt for General Evaluation of Student Answer