MSA at BEA 2025 Shared Task: Disagreement-Aware Instruction Tuning for Multi-Dimensional Evaluation of LLMs as Math Tutors*

Baraa Hikal, Mohamed Basem, Islam Oshallah, Ali Hamdi
Faculty of Computer Science, MSA University, Egypt
{baraa.moaweya, mohamed.basem1, islam.abdulhakeem, ahamdi}@msa.edu.eg

Abstract

We present MSA-MATHEVAL, our submission to the BEA 2025 Shared Task on evaluating AI tutor responses across four instructional dimensions: Mistake Identification, Mistake Location, Providing Guidance, and Actionability. Our approach uses a unified training pipeline to fine-tune a single instruction-tuned language model across all tracks, without any task-specific architecture modifications. To improve prediction reliability, we introduce a disagreement-aware ensemble inference strategy that enhances coverage of minority labels. Our system achieves strong performance across all tracks, ranking 1st in Providing Guidance, 3rd in Actionability, and 4th in both Mistake Identification and Mistake Location. These results demonstrate the effectiveness of scalable instruction tuning and disagreement-driven modeling for robust, multi-dimensional evaluation of LLMs as educational tutors.

1 Introduction

Large language models (LLMs) are increasingly used in educational applications, acting as AI tutors that engage students in natural language. However, effective tutoring goes beyond producing correct answers. AI tutors must recognize student mistakes, explain misconceptions, provide constructive guidance, and suggest actionable next steps. Assessing such teaching behavior remains challenging.

Prior work in intelligent tutoring systems (ITS) emphasized these goals long before the advent of LLMs. For example, AutoTutor used natural language processing (NLP) and dialogue-based feedback to improve learning outcomes across domains (Nye et al., 2014). Later, metrics such as *conversational uptake* were proposed to capture tutor responsiveness and its link to instructional quality (Demszky et al., 2021).

With the rise of instruction-tuned LLMs, new frameworks have emerged to assess their teaching abilities. Tack and Piech (Tack and Piech, 2022) introduced the AI Teacher Test for evaluating model helpfulness and student understanding, while later work proposed finer rubrics such as coherence, correctness, targetedness, and actionability (Macina et al., 2023; Daheim et al., 2024; Wang et al., 2024).

Building on these efforts, the BEA 2025 Shared Task adopts *MRBench*—a pedagogically motivated benchmark introduced by Maurya et al. (2025)—to evaluate AI-generated tutor responses in math dialogues (Kochmar et al., 2025). While BEA 2023 emphasized response generation, BEA 2025 shifts toward assessing feedback quality across four instructional dimensions derived from educational science.

In this work, we present MSA-MATHEVAL, a unified system that addresses all four tracks using a single fine-tuned model and consistent training pipeline. We fine-tune the open-weight `Mathstral-7B-v0.1`—an instruction-tuned LLM specialized for mathematical reasoning—using parameter-efficient LoRA adapters. To improve prediction reliability, we incorporate ensemble-based inference that combines model disagreement and uncertainty estimation.

Our contributions are as follows:

- We design a unified training pipeline for all four BEA 2025 tracks, using LoRA-based fine-tuning of `Mathstral-7B-v0.1` with no track-specific architecture changes.
- We propose an ensemble-based inference strategy leveraging model disagreement and uncertainty for robust prediction.
- We achieve top-tier performance across all tracks, including first place in Providing Guidance.

*https://github.com/baraahekal/BEA-2025

Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications, pages 1194–1202

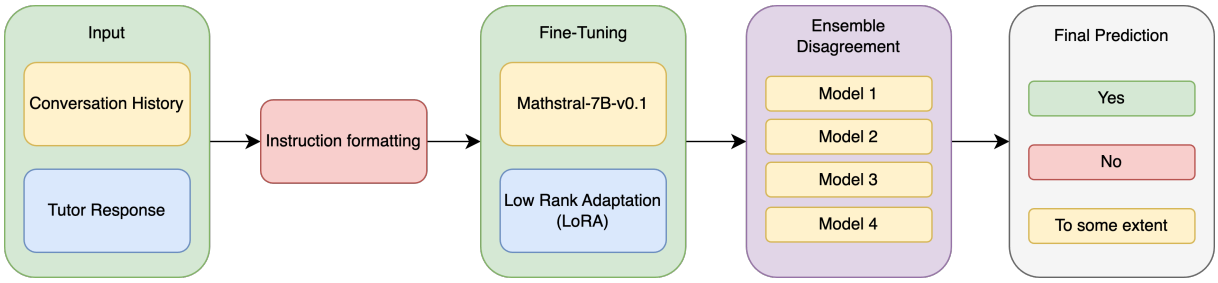

Figure 1: Overview of our unified MSA-MATHEVAL framework for the BEA 2025 Shared Task. The pipeline includes preprocessing, LoRA-based fine-tuning of `Mathstral-7B-v0.1`, and disagreement-aware ensemble inference.

2 Related Work

Evaluating the pedagogical capabilities of AI tutors builds upon long-standing research in intelligent tutoring systems (ITS) and more recent advances in large language models (LLMs). Early ITS such as AutoTutor emphasized the importance of natural language dialogue in promoting student learning through error remediation and scaffolding (Nye et al., 2014). These systems often relied on rule-based or statistical NLP methods to assess learner inputs and generate appropriate tutor responses.

The emergence of instruction-tuned LLMs has prompted a shift toward more scalable methods for modeling tutoring behavior. Tack and Piech (2022) proposed the AI Teacher Test to benchmark LLM outputs on criteria such as helpfulness and pedagogical appropriateness. Macina et al. (2023) and Daheim et al. (2024) introduced fine-grained rubrics for LLM tutoring quality in mathematical dialogue, including dimensions such as targetedness, coherence, and actionability.

In terms of modeling strategies, prior systems have explored both classification and ranking approaches for feedback generation. Daheim et al. (2024) used multi-aspect annotation schemes to evaluate feedback informativeness, while Wang et al. (2024) proposed a bridging rubric for LLM feedback grounded in human tutor behavior. These studies highlight the need for systems that go beyond correctness to capture richer instructional attributes.

Compared to these approaches, our work introduces a unified training and inference framework across multiple feedback dimensions, leveraging ensemble disagreement and uncertainty estimation for prediction stability. Unlike previous models with track-specific adaptations or rule-based post-processing, we apply a consistent architecture based on the `Mathstral-7B-v0.1` model across all tasks. This allows us to assess the generalizability of instruction-tuned LLMs for the mathematics domain across key dimensions of pedagogical ability.

3 Method

Our approach, MSA-MATHEVAL, applies a unified framework across all four tracks in the BEA 2025 Shared Task. We build on the instruction-tuned `Mathstral-7B-v0.1` model and leverage parameter-efficient fine-tuning (LoRA) along with ensemble-based inference to enhance prediction robustness. The methodology consists of the following stages: dataset preprocessing, model selection, fine-tuning strategy, and ensemble-based inference (see Figure 1).

3.1 Preprocessing

The original dataset consists of nested JSON files, where each dialogue contains multiple tutor responses annotated across four pedagogical dimensions. To facilitate instruction-based fine-tuning, we transformed the data into four track-specific JSONL files. Each file includes a flattened dialogue, a natural language evaluation prompt, and a categorical label from three possible options: *Yes*, *To some extent*, or *No*.

Each training instance was structured as a two-turn dialogue following the `chat` schema used by instruction-tuned language models. Specifically:

- **`user:`** This field contains a complete, track-specific prompt with explicit evaluation criteria, followed by the dialogue context and tutor response to be evaluated.

- **`assistant:`** This field contains the gold label corresponding to the tutor response—one of `"Yes"`, `"To some extent"`, or `"No"`—as annotated in the development set.

The system role was omitted to reduce token overhead and focus the model on the input–output mapping relevant to each multi-class classification task.

Track 1 – Mistake Identification

```
TASK DEFINITION:

You are an expert evaluator of AI tutor
responses. Your task is to determine
whether the tutor's response accurately
identifies a mistake in the student's
reasoning or solution.

EVALUATION CRITERIA:

1."Yes"- The tutor accurately identifies
a mistake in the student's response.
2."To some extent"- The tutor shows
some awareness, but it is ambiguous or
uncertain.
3."No"- The tutor fails to identify or
misunderstands the mistake.
```

Track 2 – Mistake Location

```
TASK DEFINITION:

You are an expert evaluator of AI tutor
responses. Your task is to determine
whether the tutor's response accurately
points to a genuine mistake and its
location in the student's response.

EVALUATION CRITERIA:

1."Yes"- The tutor clearly points to
the exact location of the mistake.
2."To some extent"- The tutor refers
to a mistake but is vague or indirect.
3."No"- The tutor provides no indication
of where the mistake occurred.
```

Track 3 – Providing Guidance

```
TASK DEFINITION:

You are an expert evaluator of AI tutor
responses. Your task is to determine
whether the tutor's response provides
correct and relevant guidance to help
the student.

EVALUATION CRITERIA:

1."Yes"- The tutor gives helpful
guidance such as a hint or explanation.
2."To some extent"- The guidance
is partially helpful, unclear, or
incomplete.
3."No"- The guidance is absent,
irrelevant, or factually incorrect.
```

Track 4 – Actionability

```
TASK DEFINITION:

You are an expert evaluator of AI
tutor responses. Your task is to
determine whether the tutor's feedback
is actionable, i.e., it clearly suggests
what the student should do next.

EVALUATION CRITERIA:

1."Yes"- The response includes clear
next steps for the student.
2."To some extent"- Some action is
implied, but it is not clearly stated.
3."No"- No action is suggested or the
feedback ends the conversation.
```

Each JSONL instance includes an `instruction` (as the `user` message), an `input` (composed of the full dialogue context and tutor response), and an `output` (gold label as `assistant`). This format enables effective supervised fine-tuning of `Mathstral-7B-v0.1` on each dimension-specific classification task.

3.2 Model Selection and Architecture

Our system is built upon the `Mathstral-7B-v0.1` language model, an open-source 7B-parameter transformer tailored for mathematical and scientific reasoning (Mistral AI Team, 2024). It is an instruction-tuned variant of the Mistral 7B architecture (Jiang et al., 2023), which itself builds on the transformer framework used in LLaMA (Touvron et al., 2023a,b). `Mathstral` uses a 32-layer transformer with 4096-dimensional hidden states and 32 attention heads (8 for keys/values), and benefits from Mistral's sliding-window attention mechanism, enabling long-context comprehension up to 32k tokens. This makes it particularly suitable for modeling multi-turn math tutoring dialogues that require broad context retention.

`Mathstral-7B-v0.1` was selected based on its strong performance in math-specific benchmarks and its open-access availability. It was instruction-tuned by Project Numina on mathematical reasoning tasks and achieves high scores on datasets such as GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), and MMLU-STEM (Hendrycks et al., 2021a). For instance, it reports 56.6% accuracy on MATH, significantly outperforming base Mistral and LLaMA models of comparable size.

Compared to alternatives,`Mathstral` outperforms general-purpose LLaMA 2 (Touvron et al., 2023b) and even surpasses some larger models in mathematical domains. While proprietary models like GPT-3.5 or GPT-4 (OpenAI, 2022, 2023) show impressive general capabilities, their closed nature limits fine-tuning flexibility and deployment cost-effectiveness.`Mathstral`, by contrast, is released under Apache 2.0, making it fine-tunable with LoRA on modest compute budgets.

We thus chose `Mathstral-7B-v0.1` as the backbone of our system due to its optimal trade-off

between math reasoning accuracy, open weight availability, and instruction-following capability.

3.3 Training and Fine-Tuning

We fine-tuned `Mathstral-7B-v0.1` separately for each BEA 2025 track using Low-Rank Adaptation (LoRA) (Hu et al., 2021), framing the task as three-way instruction-based classification. Each input was represented as a two-turn dialogue—comprising a prompt and a categorical label—and modeled as a supervised instruction-following task.

To enable efficient adaptation with minimal memory overhead, we used LoRA with a rank of $r = 64$, scaling factor $\alpha = 2.0$, and no dropout. Adapters were injected into the attention query and value projections in each transformer block. The low-rank update to the frozen weight matrix W is defined as:

$$\Delta W = \alpha \cdot AB \tag{1}$$

where $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times d}$ are trainable matrices, and d is the dimension of the attention head. The final effective weight is $W + \Delta W$. Figure 2 illustrates this injection mechanism.

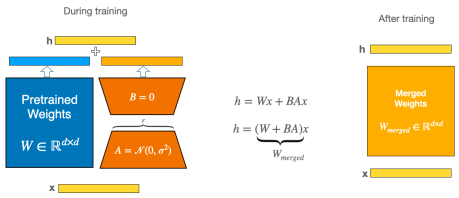

Figure 2: LoRA adaptation adds trainable low-rank matrices A and B to frozen attention weights W_0, producing an effective weight $W = W_0 + \alpha AB$ during training. Only A and B are updated, enabling memory-efficient fine-tuning (Hu et al., 2021).

Training was capped at 500 steps with gradient norm clipping ($\|g\|_2 < 1.0$) and a maximum sequence length of 2048 tokens. We used a batch size of 2, single micro-batching, and fixed seed 42 for reproducibility. Optimization was performed using AdamW with a learning rate of 4×10^{-5}, 10% linear warmup, and weight decay of 0.05.

We evaluated model performance every 50 steps on a held-out validation set, which consisted of the last 30% of the development dataset. The development set includes 300 dialogues sourced from the MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024) datasets.Checkpoints were saved every 100 steps with a retention window of the three most recent. Only LoRA adapter weights were saved to minimize disk usage and enable efficient inference. All training runs were conducted in a single-node setup with `world_size=1`.

This training configuration ensured stable convergence on limited supervision, while maintaining computational efficiency and reproducibility across all four pedagogical dimensions.

3.4 Inference and Ensemble Strategy

To enhance robustness and maintain generalization across all four tracks, we employed an ensemble-based inference strategy grounded in model disagreement. Rather than aggregating predictions through majority voting, we fine-tuned five independent models per track. Each model used the same base architecture `Mathstral-7B-v0.1` but was trained with different random seeds and shuffled data splits to encourage diversity in learned representations. This disagreement-aware mechanism allows us to capture uncertainty and preserve minority-class predictions, especially for ambiguous cases labeled `"To some extent"`.

Each model in the ensemble predicts a class independently using greedy decoding. During inference, we collect all five predictions for a given sample and apply a filtering policy: if the predictions exhibit full agreement, the class is retained. If the ensemble disagrees, we analyze the class distribution and prefer predictions that preserve the relative frequency of `"To some extent"` observed in the development set. This is crucial because `"Yes"` labels are dominant in both the training and validation sets, potentially leading to biased predictions under a naïve voting scheme.

Our design choice is motivated by the use of macro-F1 as the primary evaluation metric in the BEA 2025 Shared Task. Unlike accuracy or micro-F1, macro-F1 gives equal weight to all classes, making performance on minority labels such as `"To some extent"` especially important. By encouraging the retention of these less frequent but pedagogically relevant labels through disagreement-aware filtering, we improve per-class recall and stabilize final predictions.

This ensemble strategy is lightweight in deployment, as only LoRA adapter weights are loaded during inference. Predictions are generated sequentially and combined via a deterministic post-processing script that requires no additional training.

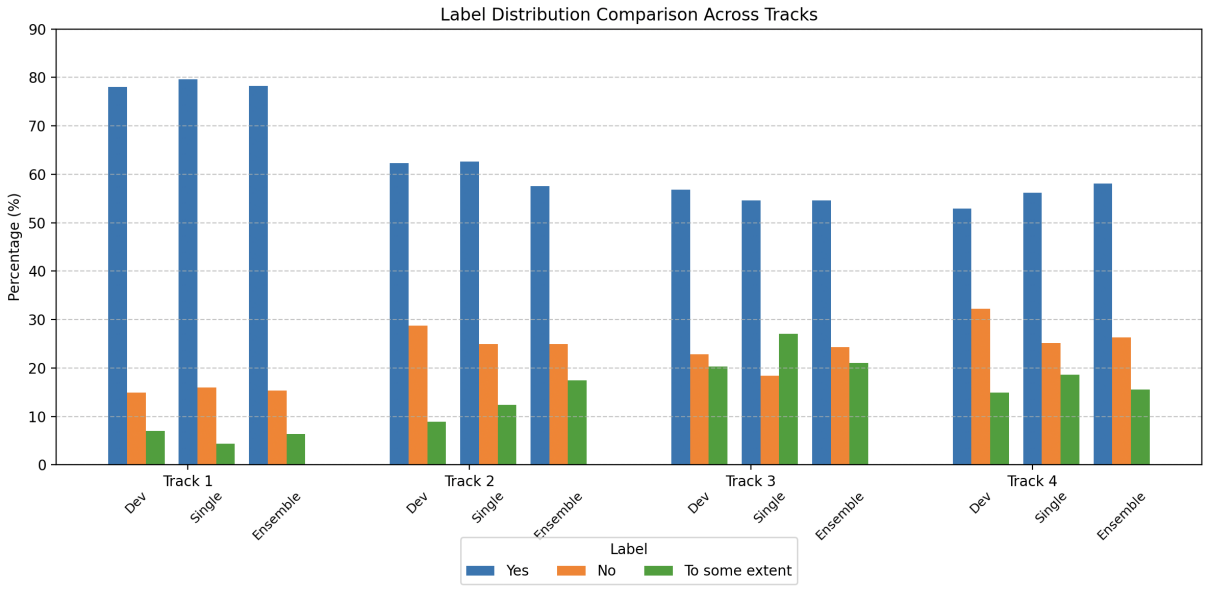

Figure 3: Label distribution comparison across four evaluation tracks. Each group of bars represents the percentage of predictions for the labels `"Yes"`, `"No"`, and `"To some extent"` for three settings: the MRBench development set (Dev), the best-performing single model on the test set (Single), and the ensemble system on the same test set (Ensemble).

4 Experiments

4.1 Dataset

The BEA 2025 Shared Task provides a benchmark for evaluating AI tutor responses across four pedagogically motivated tracks: Mistake Identification, Mistake Location, Providing Guidance, and Actionability (Kochmar et al., 2025). The dataset is based on *MRBench*, a curated collection of math-focused educational dialogues designed for evaluating feedback quality in instructional settings (Maurya et al., 2025). It includes dialogues drawn from two publicly available sources: MathDial (Macina et al., 2023) and Bridge (Wang et al., 2024).

Each instance comprises a multi-turn conversation between a student and an AI tutor, a final student question or statement, and multiple candidate tutor responses. The task is to classify each response along the four instructional dimensions, using a three-way labeling scheme: *Yes*, *To some extent*, and *No*.

The shared task organizers provide a labeled development set with expert annotations for training and validation. The test set is blind—its labels are hidden from participants and used by the organizers to evaluate final system submissions. This setup ensures fair comparison and simulates real-world deployment where labeled data may be limited or unavailable.

MRBench Statistics:

- **192** annotated dialogues in total: **60** from Bridge and **132** from MathDial.
- **1,596** total tutor responses annotated across 7 LLMs and multiple human tutors (expert and novice).
- Each response is annotated with 8 evaluation dimensions; the shared task focuses on 4 core tracks.
- **Dialogue Length:** Bridge dialogues average 4 turns and 140 characters. MathDial averages 5.5 turns and 906 characters.

4.2 Evaluation

To evaluate the pedagogical quality of model predictions across all four tracks, the BEA 2025 Shared Task employs two complementary scoring protocols: *exact evaluation* and *lenient evaluation*. Both use macro-averaged F1 score and accuracy as core metrics.

Exact Evaluation. In the primary setting, each prediction is evaluated against a gold label using a three-way classification scheme: `"Yes"`, `"To some extent"`, and `"No"`. Let C denote the set of all classes, and $F1_c$ the F1 score for class $c \in C$. The macro-F1 score is computed as the unweighted

Track	Run	Strict F1	Lenient F1	Strict Acc.	Lenient Acc.	Main Metric Rank
Mistake Identification	Run 1	**71.54%**	**91.52%**	87.59%	**95.35%**	**4th / 44**
	Run 2	70.66%	91.42%	**87.98%**	95.22%	
	Run 3	56.78%	82.95%	83.65%	91.92%	
	Run 4	67.88%	90.13%	87.20%	94.76%	
	Run 5	71.34%	**91.52%**	87.39%	**95.35%**	
Mistake Location	Run 1	55.62%	77.79%	72.01%	80.93%	**4th / 31**
	Run 2	56.02%	77.73%	**72.01%**	81.19%	
	Run 3	56.88%	**78.48%**	71.88%	**82.09%**	
	Run 4	52.79%	73.65%	63.61%	78.22%	
	Run 5	**57.43%**	**78.48%**	69.75%	**82.09%**	
Providing Guidance	Run 1	55.28%	76.02%	**67.29%**	80.35%	**1st / 35**
	Run 2	53.76%	**76.59%**	65.09%	80.74%	
	Run 3	56.65%	74.75%	63.61%	80.61%	
	Run 4	**58.33%**	77.98%	66.13%	**81.90%**	
Actionability	Run 1	51.35%	68.81%	58.31%	76.60%	**3rd / 29**
	Run 2	66.99%	84.97%	71.95%	87.91%	
	Run 3	65.90%	84.45%	71.82%	87.07%	
	Run 4	**69.84%**	**86.59%**	**75.37%**	**89.08%**	
	Run 5	65.90%	84.45%	71.82%	87.07%	

Table 1: Strict and lenient macro-F1 and accuracy across five runs per track. Bolded scores indicate per-track bests. Final column shows BEA 2025 leaderboard rank based on strict macro-F1 (main metric).

average across all classes:

$$\text{Macro-F1} = \frac{1}{|C|} \sum_{c \in C} \frac{2 \cdot \text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \quad (2)$$

This metric penalizes class imbalance and rewards systems that maintain recall across minority classes such as `"To some extent"`.

Lenient Evaluation. To account for pedagogical similarity between `"Yes"` and `"To some extent"`, the task also includes a two-way lenient evaluation protocol. Labels `"Yes"` and `"To some extent"` are merged into a single positive class, resulting in a binary classification task. The same macro-F1 and accuracy metrics are then applied to the collapsed label set.

Accuracy. For both settings, accuracy is defined as the proportion of correct predictions over all samples:

$$\text{Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = y_i) \quad (3)$$

where N is the number of samples, $\hat{y}_i$ is the predicted label, and y_i is the gold label for instance i.

Protocol. Since the test labels were not released, we computed local metrics only on the development set. All official test results were obtained through the shared task evaluation server. Model selection and early stopping were based on development macro-F1 under the exact evaluation setting, which served as the primary leaderboard metric.

4.3 Effect of Ensemble Disagreement on Label Distribution

To analyze the effect of our ensemble strategy on class balance, we examined the label distributions across all four tracks. The development set consistently exhibited a dominant proportion of `"Yes"` labels—often exceeding 55%—with `"To some extent"` and `"No"` underrepresented.

Left uncorrected, single-model predictions tended to reinforce this imbalance, frequently collapsing uncertain cases into the majority class. To mitigate this, our ensemble disagreement filtering selectively retained predictions for the minority class `"To some extent"` when model consensus was low. This design choice was informed by the use of macro-F1 as the shared task's official ranking metric, which rewards balanced performance across all classes.

Figure 3 compares label distributions from the development set, single-model outputs, and ensemble predictions. The ensemble strategy improves minority-class coverage—especially for `"To some extent"`—by better matching the development distribution and mitigating dominant-class bias. This adjustment is particularly useful in ambiguous cases where subtle feedback is warranted.

Track	Strict Macro-F1	Lenient Macro-F1	Strict Acc.	Lenient Acc.
Mistake Identification	**4th** / 44	**2nd** / 44	**1st** / 44	**2nd** / 44
Mistake Location	**4th** / 31	**6th** / 31	**10th** / 31	**6th** / 31
Providing Guidance	**1st** / 35	**2nd** / 35	**3rd** / 35	**3rd** / 35
Actionability	**3rd** / 29	**1st** / 29	**2nd** / 29	**2nd** / 29

Table 2: Per-metric leaderboard ranks (out of all teams) for each track.

5 Results

We evaluate our system across the four BEA 2025 tracks—Mistake Identification, Mistake Location, Providing Guidance, and Actionability—using both exact (three-class) and lenient (binary) evaluation protocols, as outlined in Section 4.2. We report macro-averaged F1 and accuracy scores across five independent runs for each track and compare our best results to the official leaderboard.

5.1 Performance Across Runs

Table 1 presents detailed performance scores from five independent fine-tuning runs per track. Each run was evaluated on strict and lenient macro-F1 as well as accuracy. We observe moderate variance across runs, particularly in Tracks 2 and 4, which feature more ambiguous tutor responses.

Our best-performing models achieved:

- **Track 1**: 71.54% strict macro-F1 and 91.52% lenient macro-F1 (Run 1).
- **Track 2**: 57.43% strict macro-F1 and 78.48% lenient macro-F1 (Run 5).
- **Track 3**: 58.33% strict macro-F1 and 77.98% lenient macro-F1 (Run 4).
- **Track 4**: 69.84% strict macro-F1 and 86.59% lenient macro-F1 (Run 4).

These results highlight the robustness of our unified training pipeline and the positive impact of ensemble disagreement filtering on minority-class prediction, especially in borderline cases.

5.2 Leaderboard Rankings

Table 2 summarizes our official rankings among all participating teams. We consistently placed within the top 5 across all tracks and metrics, securing the **1st** rank in Track 3 (Providing Guidance) and top-3 ranks in three other metrics.

These ranks validate the effectiveness of our approach across varied pedagogical feedback dimensions. Notably, our system generalizes well across tasks using a unified model and minimal task-specific engineering.

6 Limitations

Despite its strong performance across BEA 2025 tracks, our approach has several limitations.

First, the specialization of `Mathstral-7B-v0.1` to mathematical reasoning may hinder generalization to non-mathematical domains. While domain-specific instruction tuning improves in-domain performance, prior work has shown that such specialization can cause *catastrophic forgetting* of general knowledge, even with parameter-efficient methods like LoRA (Dettmers et al., 2023). Moreover, although LoRA significantly reduces memory and compute costs, its low-rank decomposition can constrain the model's expressiveness in capturing nuanced pedagogical feedback (Xu et al., 2023; Zhou et al., 2023).

Second, our ensemble disagreement strategy introduces additional inference cost. While it improves recall for minority labels such as `"To some extent"`, the benefit may diminish if the base models exhibit correlated predictions. Prior work shows that ensembles are most effective when model predictions are diverse and independent (Lakshminarayanan et al., 2017), which may not always hold in practice.

Finally, the reliance on macro-averaged F_1 as the primary evaluation metric, although fair for class imbalance, lacks granularity in penalizing pedagogically critical mistakes. For example, misclassifying a completely wrong tutor response as `"To some extent"` is penalized equally to a more plausible confusion between `"Yes"` and `"To some extent"`. While the lenient evaluation partially addresses this by collapsing similar labels, it does not fully capture the instructional severity of errors (Kochmar et al., 2025).

7 Conclusion

We presented MSA-MATHEVAL, a unified framework for evaluating AI tutor responses across

four pedagogical dimensions in the BEA 2025 Shared Task. By fine-tuning a math-specialized LLM (`Mathstral-7B-v0.1`) using LoRA and leveraging ensemble disagreement during inference, our system achieved top-tier results across all tracks—ranking 1st in Providing Guidance and within the top 5 in all others. Our findings highlight the effectiveness of combining domain-specific instruction tuning with disagreement-aware prediction filtering for educational feedback assessment. Future work will explore cross-domain generalization and dynamic calibration strategies to further enhance robustness.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. Stepwise verification and remediation of student reasoning errors with large language model tutors. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8386–8411, Miami, Florida, USA. Association for Computational Linguistics.

Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori Hashimoto. 2021. Measuring conversational uptake: A case study on student-teacher interactions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1638–1653, Online. Association for Computational Linguistics.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Weizhu Wang, and Zichao Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard-Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Ekaterina Kochmar, Kaushal Kumar Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Anaïs Tack, and Justin Vasselli. 2025. Findings of the BEA 2025 shared task on pedagogical ability assessment of AI-powered tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30 (NeurIPS 2017)*, pages 6402–6413.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.

Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.

Mistral AI Team. 2024. Mathstral 7b v0.1: A math reasoning and scientific discovery model. `https://mistral.ai/news/mathstral`.

Benjamin D. Nye, Arthur C. Graesser, and Xiangen Hu. 2014. Autotutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, 24(4):427–469.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. `https://openai.com/blog/chatgpt`.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anaïs Tack and Chris Piech. 2022. The AI teacher test: Measuring the pedagogical ability of blender and GPT-3 in educational dialogues. In *Proceedings of the 15th International Conference on Educational*

Data Mining, pages 522–529, Durham, United Kingdom. International Educational Data Mining Society.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 2174–2199, Mexico City, Mexico. Association for Computational Linguistics.

Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. 2023. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. 2023. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *arXiv preprint arXiv:2303.07338*.