Trojsten Benchmark: Evaluating LLM Problem-Solving in Slovak STEM Competition Problems

Adam Zahradník[1,2] and **Marek Šuppa**[1,2,3]

[1] Faculty of Mathematics, Physics and Informatics, Comenius University in Bratislava
[2] NaiveNeuron [3] Cisco Systems

adam@zahradnik.xyz marek.suppa@fmph.uniba.sk

Abstract

Large language models show promising performance on reasoning tasks, yet evaluation methods for low-resource languages remain limited, particularly for complex STEM problem-solving. We introduce Trojsten Benchmark, a Slovak-language dataset of 1,108 high-school competition problems with reference solutions across mathematics, physics, and programming, and a rubric-based LLM grading framework. Using GPT-4 to generate rubrics and grade solutions, we observe 1.05 average absolute deviation from human graders (5-point scale), while benchmarking GPT-3.5-Turbo, GPT-4, GPT-4o, and open-weight models (Llama 3, Phi-3). We quantify multistep reasoning performance by difficulty, show consistent underperformance on harder items, and demonstrate language sensitivity: accuracy drops on English translations of Slovak statements, evidencing challenges beyond translation. Trojsten Benchmark complements English-centric math datasets (e.g., MATH, GSM8K) by targeting open-response, rubric-gradable reasoning under low-resource linguistic framing. We release code and data to enable reproducible evaluation and human-aligned auto-grading for STEM in under-served languages.

1 Introduction

The growing capabilities of Large Language Models (LLMs) have transformed the landscape of educational assessment (Li et al., 2023; Wang et al., 2024; Gan et al., 2023; Kasneci et al., 2023; Phung et al., 2023). Although previous experiments reported LLMs that have been pre-trained (Wu et al., 2023) or fine-tuned to provide grading labels (Latif and Zhai, 2024; Organisciak et al., 2023), LLMs are increasingly applied in grading open-ended responses and providing feedback beyond traditional methods that rely on exact answers (e.g., multiple-choice or numerical questions) (Chang and Ginter, 2024; Divya et al., 2023; Fagbohun et al., 2024; Koutcheme et al., 2024). However, previous work largely focuses on shorter, simpler answers, without addressing the more demanding multistep reasoning often required in STEM subjects (Yan et al., 2024).

Historically, human evaluators have set the benchmark for grading complex, descriptive, and problem-solving tasks. In this work, we introduce the first dataset centered on Slovak high-school competition problems in mathematics, physics, and programming, a domain where datasets are scarce. These problems present a unique challenge: while publicly available in PDF form, their formatting makes them difficult to process automatically, ensuring that they are unlikely to have been included in large-scale LLM training datasets. This allows for a relatively unbiased evaluation, providing a more accurate reflection of LLM performance compared to datasets in widely represented languages.

To the best of our knowledge, this is the first work that focuses on complex STEM problems requiring multistep reasoning while involving human experts to evaluate both generated rubrics and the evaluations based on those rubrics. Previous studies, such as (Wu et al., 2024), (Sawada et al., 2023) and (Chiang et al., 2024), rely on expert-provided rubrics, model-generated rubrics or short-answer assessments, respectively, without expert verification or a focus on multistep problems. Moreover, perhaps the closest work to ours (Xie et al., 2024), explores LLM-based grading in an Operating Systems course using the Mohler dataset (Mohler and Mihalcea, 2009; Mohler et al., 2011), but does not employ human evaluators to the same extent nor targets complex problem-solving tasks.

While the formal principles of mathematics, physics, and programming are universal, their expression in natural language – particularly in complex, multi-step word problems – is not. Datasets from high-school competitions, such as the Trojsten Benchmark, are developed within a specific

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 35095–35110

educational tradition and cultural context. This results in unique linguistic framing, idiomatic phrasing, and problem narratives that may differ significantly from the English-centric corpora on which most large language models are trained. A central goal of our work is therefore to investigate whether this linguistic specificity provides a genuine challenge beyond simple translation effects. Indeed, our experiments confirm this hypothesis, revealing that model performance is highly sensitive to the source language. For instance, we found that GPT-4's performance on a subset of our problems dropped from an average score of 3.30 in the original Slovak to just 1.32 when manually translated into English (Section 5.4), demonstrating that the original low-resource formulation presents a distinct and non-trivial reasoning challenge.

Our contributions are fourfold: (1) We present the first comprehensive dataset of Slovak competition problems in mathematics, physics, and programming, with reference solutions, tailored to support educational research in low-resource languages; (2) We introduce an LLM-based grading framework that leverages GPT-4 (OpenAI et al., 2024) to generate detailed rubrics and evaluate student solutions, aiming to address the complexities of STEM problem-solving; (3) We systematically compare GPT-4's grading performance with human evaluations, emphasizing the gaps that still exist, particularly in nuanced or complex responses; and (4) We introduce baseline benchmarks on our dataset using various large language models and prompting techniques, providing insight into their reasoning abilities in Slovak language.

The benchmark code and data used for experiments in this paper are published on GitHub[1].

1.1 Related work

Large language models have already been evaluated on mathematical reasoning tasks by researchers using numerous datasets, most of which were created by scraping problems from the internet or standardized tests. We provide a comparison of a selection of datasets related to our work.

MATH is a dataset consisting of challenging competition mathematic problems with step-by-step natural language solutions introduced by Hendrycks et al. (2021). The problems were retrieved from United States' mathematics competitions. *GSM8K* released by Cobbe et al. (2021) consists of multistep elementary school word problems with natural language solutions. *MGSM* is a multilingual dataset introduced by Shi et al. (2022) containing 250 manually translated grade-school problems from the GSM8K. Various other datasets such as *Omni-MATH* (Gao et al., 2024), *CHAMP* (Mao et al., 2024) and *MathOdyssey* (Fang et al., 2024) explore similar types of problems in the high-school competition space, or harder.

Despite these advancements, there remains a scarcity of datasets and evaluation methods for complex STEM problems in lower-resource languages. To the best of our knowledge, our work introduces the first comprehensive dataset of Slovak competition problems in mathematics, physics, and programming designed to evaluate the reasoning capabilities of LLMs on authentic problem sets in a low-resource language.

2 Problem dataset

The problem dataset used in this paper contains various problems and their solutions from Slovak high-school competitions. As they are competition problems, they are designed to be challenging for high-school students. These problems are designed so that an average student should be able to solve about half of them. The dataset contains problems from three categories: maths, physics and programming. These problems usually require the student to embrace innovative approaches, and to document them thoughtfully in their solution.

2.1 Creating the dataset

The dataset was sourced from competition archives, originally available as Markdown or LaTeX documents.

Each problem was manually reviewed by a human annotator for classification, to ensure overall quality, and to filter out items that lacked sufficient information in the problem statement itself (e.g., problems requiring videos or linking to external websites).

The final dataset consists of 1,108 problems and their solutions: 361 from mathematics, 479 from physics, and 268 from programming competitions. The materials span approximately eight years of national-level STEM competitions.

This dataset is comparable in size and scope to the one introduced in Sawada et al. (2023). However, unlike that dataset—which includes a substantial number of multiple-choice problems—all

[1] https://github.com/gardenerik/trojsten-benchmark

problems in our dataset are open-ended, without any answer options provided.

2.2 Overview of the problems

The problems in our dataset are not only diverse in terms of the primary target area (maths, physics and algorithms), but in addition, there are also different types of such problems.

2.2.1 Maths problems

The overwhelming majority (189 problems) of our maths problems are based on the student having to prove whether a given statement is true or not. Other problems require the student to quantify some equations or otherwise calculate a numerical result (84 problems). There are some (42) problems that want to enumerate all numbers, functions, etc. that satisfy certain conditions. Furthermore, there happens to be a tiny number (4) of problems that require to carry out some geometric construction. An example maths problem is provided in Figure 1.

> Let $f : \mathbf{R}^+ \to \mathbf{R}$ be a function such that the functions $f(x) - x^3$ and $f(x) - 3x$ are increasing. Determine whether the function $f(x) - x^2 - x$ must be monotone.

Figure 1: A sample problem from the "maths" part of our dataset. The problem text was translated to English for consistency.

2.2.2 Physics problems

We divide physics problems into two categories. The first category of problems is problems that only need theoretical knowledge to explain a relationship between physics variables or explain some physical phenomena. They usually involve figuring out some equations, explaining them, and using them to obtain an answer to the question. This category makes up 428 of the problems. The second category requires the student to come up with an experiment setup, execute and document the experiment. There are 51 such problems. An example problem is provided in Figure 2.

2.2.3 Algorithmic problems

All of our algorithmic problems focus on figuring an effective way to solve some problem. This usually means using different algorithms in unusual ways or coming up with new algorithms to solve the problem. Algorithmic problems are also among the longest in our dataset. This is because they contain plenty of details about the input and output format, input size constraints, along with a fictional story to provide some practical background to the problem. An excerpt from one such problem is provided in Figure 3.

> I'm sitting in a bubble bath and bubbles are flowing up my back. They seem very cold, perhaps even colder than the surrounding air. Why is that?

Figure 2: A sample problem from the "physics" part of our dataset. The problem text was translated to English for consistency.

> You have been given points on a plane. Find out how non-random they are, that is, the vertices of how many triangles they form.

Figure 3: A sample problem (shortened into an excerpt) from the "algorithmic" part of our dataset. The problem text was translated to English for consistency.

2.3 Difficulty

The problems in our dataset have varying degrees of difficulty. In the real competitions they were taken from, they tend to be sorted by estimated difficulty. The few first problems should be solvable by all high school students, whereas the last problems are usually solved only by students engaging in national or international competitions. The relative difficulty data is used to assign every problem a difficulty score on a scale of 1 to 10, with 10 being the most difficult. The difficulty distribution is shown in Figure 4. This score will later be used to quantify the abilities of large language models in solving these problems. Some of our competitions did have less than 10 problems in one round, in which case we distributed their difficulty evenly across the 1-10 scale.

2.4 Length

An average problem in our dataset has 195 words. However, most of our problems are less than 200 words long, as shown on Figure 5. This is because mathematical and physical problems are typically brief, whereas algorithmic problems tend to be more extensive. This anomaly was discussed in

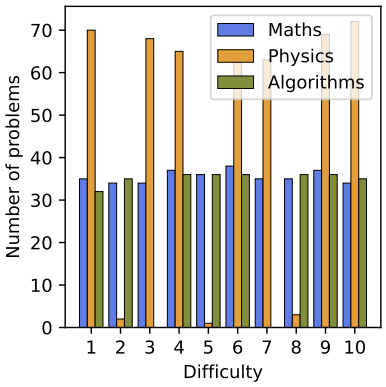


Figure 4: Problem difficulty distribution across the whole dataset.

Section 2.2.3, and is mostly due to a longer problem story and details about handling input and output data.

2.5 Overview of the solutions

As stated previously, our dataset contains solutions in natural language for every problem. That means that the solutions contain explanations, proofs and other details. An example of such a natural language solution is provided in Figure 6.

Our reference solutions vary greatly in their word count. An average reference solution is 652 words long, with the longest reference solution consisting of 3,682 words. In our maths problems, the average length is 487 words. For our physics problems, the typical length increases to 660 words. Meanwhile, our algorithmic problems tend to have more detailed solutions, with an average length of 850 words. The distribution of the word count is shown in Figure 7.

3 Grading method

The most straightforward way to grade a solution using LLM is to provide the model with the reference solution and the answer that should be evaluated. It is then asked to grade the provided answer. Prior research indicates that such an approach is possible, but the evaluations are not reliable enough to be used alone (Kortemeyer, 2023b; Schneider et al., 2023). Some researchers went so far as to avoid providing the model with the reference solution. We have experimented zero-shot prompting the model with the reference solution and the student's solution. Our results during preliminary experiments were unreliable and inconsistent, similar to those observed by Kortemeyer (2023a) in a similar experiment. This has motivated us to focus on other methods.

An improved approach was introduced by asking the model to generate evaluation rubrics, and then using those rubrics to evaluate the solutions (Sawada et al., 2023). The model is provided with the reference solution and generates rubrics and allocates points to them. It was shown by Sawada et al. (2023) that GPT-4 designs rubrics that cover most of the solution steps correctly, but sometimes fail to properly allocate points based on their importance. The authors further discovered that the model is quite reliable on assigning the correct number of points to solutions based on the generated rubrics. However, the model cannot score solutions that do not follow the generated rubrics, but are otherwise correct. Another issue with this approach is that the model attempts to assign points to attempted solutions that are outside the generated rubrics. A human evaluator would score these solutions with zero points.

Aware of its limitations, our approach to evaluating the answers is inspired by the work of Sawada et al. (2023). We zero-shot GPT-4 with the reference solution and prompted it to generate a grading rubric. The used prompts are outlined in Appendix D.

Then, those generated rubrics are used to produce a grading score on a scale of 0 to 10. This is achieved by zero-shot prompting the model with the rubric and the student's solution. We also asked the model to provide a comment for every point in the rubric, as this improved its consistency. The model concluded its output with the final score. Our method is also visualized in Figure 8.

After some experiments, we have established the GPT-4 model as our grader and rubric generator. At first, we tried using GPT-3.5-Turbo, but were dissatisfied with its capabilities. The model was often referencing to non-existent claims in the solution or the grading rubric itself. It sometimes made entirely new and incorrect claims about the concepts involved in the problem. The model also failed to keep attention to details, which was most noticed in maths expressions. For example, the model did not notice a difference between $\frac{x}{2}$ and $\frac{x+1}{2}$. We also found that the model failed to follow the mathematical reasoning of a solution properly, probably due to the aforementioned issues. We then experimented with the larger GPT-4 model, with which we did not experience most of those problems. We also noticed that the quality of produced comments was greatly improved.

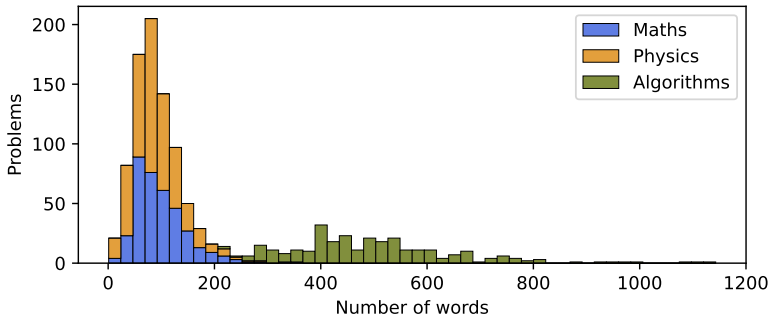


Figure 5: Number of words in the problem statements that can be found in the dataset.

> When we put our finger below the surface, the water level rises a little. This will increase the hydrostatic pressure at the bottom of the right bowl. Since the pressure at the bottom of the left bowl has not increased, the scales will tip to the right.

Figure 6: A sample solution that can be found in the dataset. The solution text was translated to English for consistency.

All grading experiments were run against GPT-4 using the Azure OpenAI endpoints [2] using the API version `2023-12-01-preview` with the temperature set to 0 to aid reproducibility. An extended discussion on the resource requirements in terms of utilized tokens as well as the full cost of the experiments can be found in Appendix C. GPT-4 was used because it had the best performance at the time of design.

We later also experimented with Llama3 as a grader. The obtained results indicate that, despite lower absolute scores, Llama preserves the relative rankings while also reporting robust Pearson and Spearman correlations. Detailed results are attached in Appendix B. Taken together, we believe this suggests that it can serve as a GPT-4 replacement when access to a proprietary model may be problematic.

4 Evaluation

We have executed various experiments to evaluate the quality of rubrics and grading produced by LLMs and compare them with those provided by five volunteer human evaluators. Each of them had prior experience with evaluation of the respective competition problems, and each problem was evaluated by at least two evaluators. All five human evaluators were Slovak university students.

This was done on various problems from our local maths, physics and computer science competitions for high school students. We have tried our best to select problems with various difficulty levels and to keep their selection balanced.

4.1 Consistency with reference solutions

As a sanity check, our method was tested by providing GPT-4 with the reference solutions of 112 problems to grade them. On average, GPT-4 graded the reference solution with 9.6 points out of 10, with most of the scores being 10/10, as shown in Figure 9.

4.2 Quality of the generated rubrics

We randomly picked a subset of the generated rubrics (20 rubrics per subset, 60 in total) and asked the competition organizers whether they agreed with them by using a 5-point Likert scale[3]. By doing this, we gained insight into the quality of the rubrics themselves. On average, our organizers reached an agreement Likert score of 3.98 with a median of 4, a standard deviation of 0.93 and Cohen's kappa inter-annotator agreement score of 0.24, which suggests fair agreement as outlined by Landis and Koch (1977).

4.3 Ability to correctly assign points

We asked the organizers to verify GPT-4's scoring against the rubric by manually grading two randomly selected generated solutions for each evaluated rubric (around 120 solutions in total). This was done to assess GPT-4's ability to follow the rubric

[2] https://learn.microsoft.com/en-us/azure/ai-services/openai/reference

[3] The Likert items used were: 1 Strongly disagree / 2 Disagree / 3 Neither agree nor disagree / 4 Agree / 5 Strongly agree

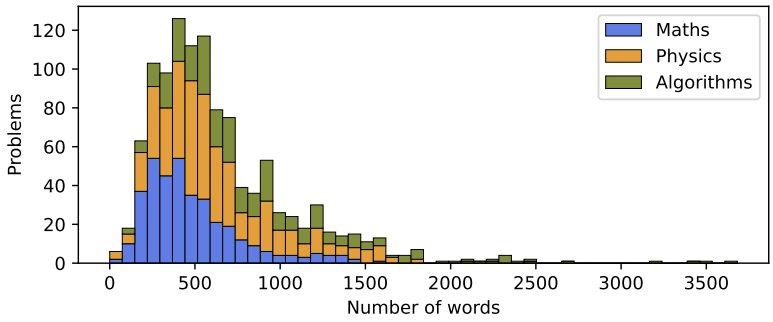


Figure 7: Number of words in the reference solutions that can be found in the dataset.

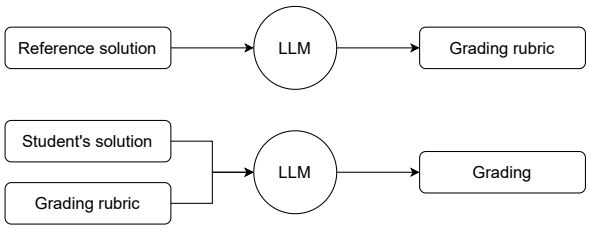


Figure 8: A visualization of our grading method in the form of a diagram.

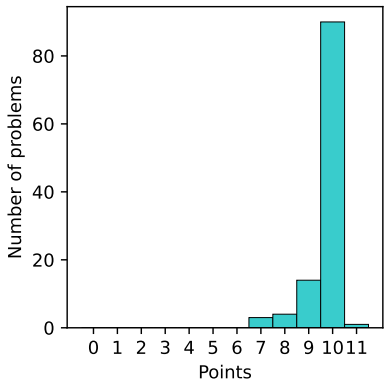


Figure 9: Scores awarded by GPT-4 to the sample reference solutions.

and accurately assign and tally points. The average absolute difference between GPT-4's grading and that of a human evaluator, when both followed the rubric, was 1.05 points. On average, GPT-4 assigned 0.9 points more than it should have, indicating a consistent overestimation. The comparison between the points awarded by GPT-4 and the human evaluators is shown in Figure 10.

4.4 Comparing to human evaluators

We also asked our organizers to grade the solutions that we provided them in the section 4.3 as if they were grading the problem themselves, without using the rubric. This provides an additional insight into the quality of both rubrics and grading by the model. The average absolute difference

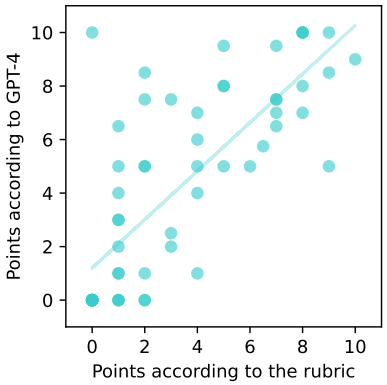


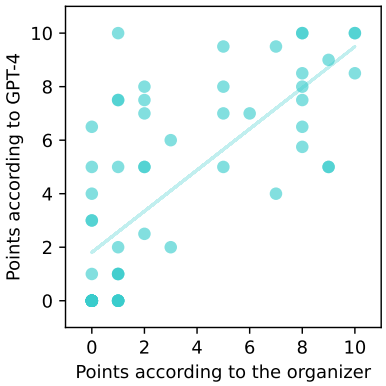
Figure 10: Relationship between GPT-4 and human evaluator scores when following the rubric ($R^2 = 0.5948$)

between points awarded by GPT-4 and the organizers was 1.87. Additionally, GPT-4 awarded on average 1.11 more points than the human evaluator. The relationship between these scores is shown in Figure 11a. Cohen's kappa agreement score between human evaluators was 0.35, suggesting fair agreement as per Landis and Koch (1977).

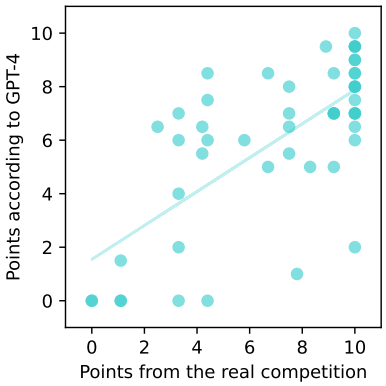
Additionally, we analyzed 50 graded students' solutions from the competition (i.e. solutions that were submitted irrespective of our experiments) for comparison with GPT-4's grading. These solutions were graded by a human evaluator during the competition, so they are graded more consistently than the previous experiment. GPT-4 did provide a score 1.1 points higher than the organizer, with an average absolute difference of 2.2 points. Figure 11b shows the relation between points scored during the competition and points scored according to the GPT-4 model.

4.5 Rubrics error analysis

We also discussed the challenges human evaluators encounter when working with LLM-generated rubrics and grading. One issue they identified was that when a reference solution contained multi-



(a) Points awarded by GPT-4 vs. human evaluator ($R^2 = 0.5226$)



(b) Points awarded by GPT-4 vs. human evaluator during competition ($R^2 = 0.4770$)

Figure 11: Comparison of points awarded by GPT-4 and human evaluators

ple correct approaches (e.g., different methods or efficiencies), the generated rubric often expected the student to include all of them in their answer (in 9.7% of rubrics). This could be addressed by improving the prompts or refining the reference solutions beforehand. Another problem was that GPT-4 occasionally confused mathematical expressions. In one case, GPT-4 refused to recognize that $L = R$ and $R = L$ are equivalent. Minor issues were present in 45.2% of rubrics, where the model incorrectly copied equations from the solution or introduced minor errors. This could potentially be mitigated by allowing the LLM to use external tools to verify symbolic relationships. Evaluators also noted that GPT-4 sometimes overlooked important details in the student's reasoning that they would have flagged and deducted points for. In 6.5% of the rubrics, GPT-4 failed to award points for alternative correct solutions. Another 6.5% of rubrics were too broad, allowing incomplete or incorrect solutions to receive high scores.

5 Benchmarking existing models

We then continued to benchmark existing large language models on our dataset using the rubric-based evaluation described in Section 3.

We have run the benchmark against GPT-3.5-Turbo, GPT-4, GPT-4o and also open-weight models Llama 3 (70B), Phi 3 (mini) and Phi 3 (medium). In all tests, grading was done by GPT-4.

We have also tested different prompting techniques to compare their influence on the reasoning in the Slovak language. Our tests include zero-shot prompting, few-shot prompting (Brown et al., 2020), zero-shot chain-of-thought (Kojima et al., 2023), generated knowledge and dual-prompt generated knowledge (Liu et al., 2022).

Overall, we were able to achieve best scores of 2.89 (GPT-3.5-Turbo), 4.70 (GPT-4), 6.07 (GPT-4o), 3.83 (Llama 3 70B), 1.30 (Phi 3 Mini) and 2.72 (Phi 3 Medium). Models achieved the worst scores in our maths subset and best in our algorithms subset.

5.1 Prompting techniques

We have used various prompting techniques to measure the LLM's capabilities. In large commercial models, the greatest increase in points scored can be achieved by using one of the generated knowledge approaches. By using GK, we have measured an increase from the zero-shot average score of 6.14 to 8.51 points in our algorithms subset for GPT-4o. It should be noted, however, that on our maths problems, using the generated knowledge approach results in worse scores for both GPT-3.5-Turbo and GPT-4.

In smaller, open-weight models, the pattern is similar, with Phi 3 Medium improving from 2.78 points zero-shot average to 4.78 on our algorithms subset. Llama 3 70B was also able to improve its score by employing generated knowledge in our maths subset, going from 2.95 points on average to 5.51. For complete results, see Appendix A.

5.2 Effect of problem difficulty

A pattern similar to our real competition data appears when measuring scores achieved by the LLMs relative to the problem's difficulty. The language models struggle to score points as the problem difficulty increases, as shown in Figure 12.

Even when the model scores a high number of points on average, it still scores fewer points the

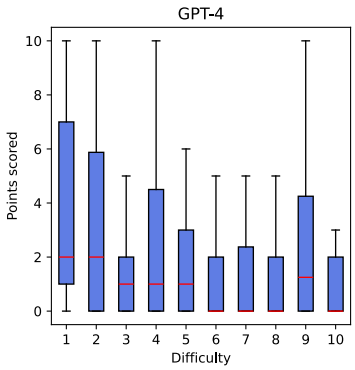


Figure 12: GPT-4's achieved scores vs. problem difficulty on maths subset

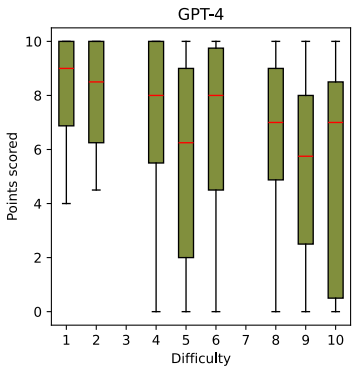


Figure 13: GPT-4's achieved scores vs. problem difficulty on algorithms subset

more difficult the problem gets, which is indicated in Figure 13. This is consistent with score vs. difficulty distribution observed on real competition participants.

The models do not get better consistently on the whole difficulty range, however. The highest increase in points scored can be seen in the least difficult problems, whereas the most difficult problems exhibit the smallest improvement.

5.3 Problem solution language

Figure 14 shows the difference between scores obtained when the model generated its solution in English and Slovak. In zero-shot experiments, the models decided to output English solutions even though the problems were in Slovak in 71.3% of cases, with GPT-3.5-Turbo preferring English output more (89.2% of cases), GPT-4 preferring Slovak (only 39% of solutions were in English) and GPT-4o preferring English (98.8% of solutions). The models achieved mean score of 3.53 when outputting Slovak and 4.14 when outputting English.

When we look at the results per model, we see

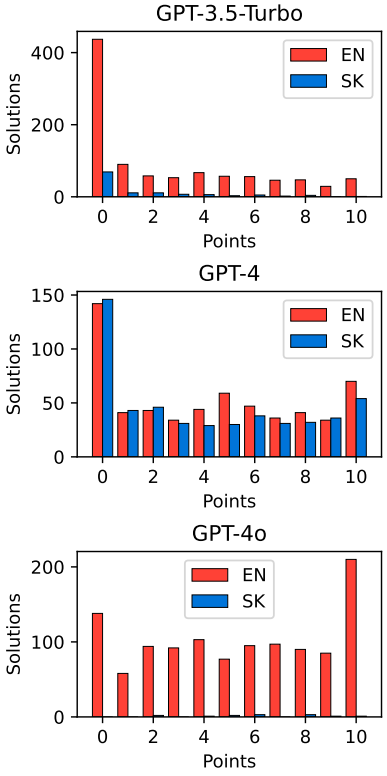


Figure 14: Models' performance when outputting text in Slovak and English

that GPT-3.5-Turbo scores on average 1.47 in Slovak and 2.69 in English. GPT-4 also excels in English solutions by scoring 4.27 on average, while scoring an average of 3.95 in Slovak. GPT-4o achieved similar results in both languages, scoring 5.96 in Slovak and 5.33 in English. These results are shown in Figure 14. The tested open weight models always produced output in English, so we will not compare theirs scores.

We also checked that the difficulty of problems was distributed almost evenly across both languages.

5.4 Problem statement language

We have also experimented with translating the problem statements into English. A small subset of our maths problems ($n = 36$) was hand-translated into English and prompted to the models.

In GPT-3.5-Turbo, there almost was no measurable difference between the scores received from English and Slovak statements. On Slovak statements, GPT-3.5-Turbo scored on average 0.60, while it scored 0.58 on English statements.

When tested with GPT-4, the average score for Slovak statement was 3.30 and 1.32 for English.

6 Conclusion

This paper introduced the Trojsten Benchmark, a novel dataset comprising 1,108 high-school STEM competition problems and their solutions in Slovak, a lower-resource language. We developed and validated an LLM-powered, rubric-based grading methodology using GPT-4 for both rubric generation and solution evaluation. Our experiments demonstrated that GPT-4 can generate evaluation rubrics with which human competition organizers achieved a mean agreement Likert score of 3.98 out of 5. When these generated rubrics were employed for grading, GPT-4's scores exhibited an average absolute difference of only 1.05 points compared to human evaluators, although GPT-4 tended to overscore by an average of 0.9 points.

Utilizing this dataset and our grading framework, we conducted extensive benchmarking of several large language models, including GPT-3.5-Turbo, GPT-4, GPT-4o, Llama 3 70B, Phi 3 Mini, and Phi 3 Medium, across various prompting techniques. Our findings reveal that contemporary LLMs possess promising, albeit still developing, reasoning capabilities in Slovak for complex STEM problems, with GPT-4o achieving the highest average score of 6.07 out of 10. We observed a consistent trend where model performance decreased with increasing problem difficulty, mirroring patterns seen in human participants. Notably, translating problem statements from Slovak to English did not uniformly enhance performance; for instance, GPT-4 performed better on problems presented in Slovak.

7 Limitations

Our work introduced here has several constraints that require future investigation:

The rubric generation process exhibits GPT-4's hallucinations of mathematical relationships. The rubric grading process shows similar problems, potentially penalizes different, but valid approaches that were not explicitly stated in the reference solution.

Our benchmarking method and its evaluation, while sufficient for preliminary benchmarking, does not fully account for error propagation across the various stages (rubric generation, grading, ...).

Our benchmarking method does not fully account for error propagation across stages. Furthermore, biases in LLM training data may affect performance on Slovak-specific nuances. Our translation experiments confirm these nuances are impactful, as model performance was not consistently robust across languages and, in the case of GPT-4, was significantly higher on the original Slovak problems.

Even though our work shows promising results in solution grading by LLMs, such systems should only be used as a hint for human evaluators, due to their unreliability and various problems discussed. Exclusive use of LLMs in grading applications poses serious risks and should be discouraged.

Acknowledgments

This work was partially funded by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I02-03-V01-00029 and by grant APVV-21-0114.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

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

Cheng-Han Chiang, Wei-Chih Chen, Chun-Yi Kuan, Chienchou Yang, and Hung-yi Lee. 2024. Large language model as an assignment evaluator: Insights, feedback, and challenges in a 1000+ student course. *arXiv preprint arXiv:2407.05216*.

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. *Preprint*, arXiv:2110.14168.

Arunima Divya, Vivek Haridas, and Jayasree Narayanan. 2023. Automation of short answer grading techniques: Comparative study using deep learning techniques. In *2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–7. IEEE.

O Fagbohun, NP Iduwe, M Abdullahi, A Ifaturoti, and OM Nwanna. 2024. Beyond traditional assessment:

Exploring the impact of large language models on grading practices. *Journal of Artifical Intelligence and Machine Learning & Data Science*, 2(1):1–8.

Meng Fang, Xiangpeng Wan, Fei Lu, Fei Xing, and Kai Zou. 2024. Mathodyssey: Benchmarking mathematical problem-solving skills in large language models using odyssey math data. *Preprint*, arXiv:2406.18321.

Wensheng Gan, Zhenlian Qi, Jiayang Wu, and Chun-Wei Lin. 2023. Large language models in education: Vision and opportunities. *2023 IEEE International Conference on Big Data (BigData)*, pages 4776–4785.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *Preprint*, arXiv:2410.07985.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. *Preprint*, arXiv:2103.03874.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, George Louis Groh, Stephan Günnemann, Eyke Hüllermeier, Stephan Krusche, Gitta Kutyniok, Tilman Michaeli, Claudia Nerdel, Jürgen Pfeffer, Oleksandra Poquet, Michael Sailer, Albrecht Schmidt, Tina Seidel, Matthias Stadler, Jochen Weller, Jochen Kuhn, and Gjergji Kasneci. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Gerd Kortemeyer. 2023a. Performance of the pre-trained large language model GPT-4 on automated short answer grading. *Preprint*, arXiv:2309.09338.

Gerd Kortemeyer. 2023b. Toward AI grading of student problem solutions in introductory physics: A feasibility study. *Physical Review Physics Education Research*, 19(2).

Charles Koutcheme, Nicola Dainese, Arto Hellas, Sami Sarsa, Juho Leinonen, Syed Ashraf, and Paul Denny. 2024. Evaluating language models for generating and judging programming feedback. *arXiv preprint arXiv:2407.04873*.

J Richard Landis and Gary G Koch. 1977. An application of hierarchical kappa-type statistics in the assessment of majority agreement among multiple observers. *Biometrics*, pages 363–374.

Ehsan Latif and Xiaoming Zhai. 2024. Fine-tuning chatgpt for automatic scoring. *Computers and Education: Artificial Intelligence*, 6:100210.

Qingyao Li, Lingyue Fu, Weiming Zhang, Xianyu Chen, Jingwei Yu, Wei Xia, Weinan Zhang, Ruiming Tang, and Yong Yu. 2023. Adapting large language models for education: Foundational capabilities, potentials, and challenges. *arXiv preprint arXiv:2401.08664*.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. 2022. Generated knowledge prompting for commonsense reasoning. *Preprint*, arXiv:2110.08387.

Yujun Mao, Yoon Kim, and Yilun Zhou. 2024. Champ: A competition-level dataset for fine-grained analyses of llms' mathematical reasoning capabilities. *Preprint*, arXiv:2401.06961.

Michael Mohler, Razvan C. Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Annual Meeting of the Association for Computational Linguistics*.

Michael Mohler and Rada Mihalcea. 2009. Text-to-text semantic similarity for automatic short answer grading. In *Conference of the European Chapter of the Association for Computational Linguistics*.

OpenAI, Josh Achiam, and Steven Adler et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Peter Organisciak, Selcuk Acar, Denis Dumas, and Kelly Berthiaume. 2023. Beyond semantic distance: Automated scoring of divergent thinking greatly improves with large language models. *Thinking Skills and Creativity*, 49:101356.

Tung Phung, Victor-Alexandru Pădurean, José Pablo Cambronero, Sumit Gulwani, Tobias Kohn, Rupak Majumdar, Adish Kumar Singla, and Gustavo Soares. 2023. Generative ai for programming education: Benchmarking chatgpt, gpt-4, and human tutors. *Proceedings of the 2023 ACM Conference on International Computing Education Research - Volume 2*.

Tomohiro Sawada, Daniel Paleka, Alexander Havrilla, Pranav Tadepalli, Paula Vidas, Alexander Kranias, John J. Nay, Kshitij Gupta, and Aran Komatsuzaki. 2023. ARB: Advanced reasoning benchmark for large language models. *Preprint*, arXiv:2307.13692.

Johannes Schneider, Bernd Schenk, Christina Niklaus, and Michaelis Vlachos. 2023. Towards LLM-based autograding for short textual answers. *Preprint*, arXiv:2309.11508.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners. *Preprint*, arXiv:2210.03057.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024. Large language models for education: A survey and outlook. *ArXiv*, abs/2403.18105.

Xuansheng Wu, Xinyu He, Tianming Liu, Ninghao Liu, and Xiaoming Zhai. 2023. Matching exemplar as next sentence prediction (mensp): Zero-shot prompt learning for automatic scoring in science education. In *International conference on artificial intelligence in education*, pages 401–413. Springer.

Xuansheng Wu, Padmaja Pravin Saraf, Gyeong-Geon Lee, Ehsan Latif, Ninghao Liu, and Xiaoming Zhai. 2024. Unveiling scoring processes: Dissecting the differences between llms and human graders in automatic scoring. *arXiv preprint arXiv:2407.18328*.

Wenjing Xie, Juxin Niu, Chun Jason Xue, and Nan Guan. 2024. Grade like a human: Rethinking automated assessment with large language models. *arXiv preprint arXiv:2405.19694*.

Lixiang Yan, Lele Sha, Linxuan Zhao, Yuheng Li, Roberto Martinez-Maldonado, Guanliang Chen, Xinyu Li, Yueqiao Jin, and Dragan Gašević. 2024. Practical and ethical challenges of large language models in education: A systematic scoping review. *British Journal of Educational Technology*, 55(1):90–112.

A Benchmark results for various LLMs

Subset	Approach	GPT-3.5-Turbo	GPT-4	GPT-4o
Maths	Zero-Shot	1.24	2.26	3.36
	One-Shot	**1.36**	**2.67**	-
	Zero-Shot CoT	1.00	2.08	3.37
	Gen. Knowledge	1.15	2.31	3.24
	Dual-Prompt GK	1.16	2.25	**3.39**
Physics	Zero-Shot	2.46	4.15	6.06
	One-Shot	2.30	4.38	-
	Zero-Shot CoT	2.49	**4.51**	6.12
	Gen. Knowledge	2.36	4.18	6.20
	Dual-Prompt GK	**2.66**	4.43	**6.31**
Algorithms	Zero-Shot	4.35	6.38	6.14
	One-Shot	4.15	3.19	-
	Zero-Shot CoT	4.64	6.44	7.92
	Gen. Knowledge	4.57	5.36	**8.51**
	Dual-Prompt GK	**4.66**	**6.91**	8.24

Subset	Approach	Llama 3 70B	Phi 3 Mini	Phi 3 Medium
Maths	Zero-Shot	1.86	0.46	0.81
	One-Shot	-	-	-
	Zero-Shot CoT	1.82	**0.57**	0.99
	Gen. Knowledge	1.83	0.43	**1.03**
	Dual-Prompt GK	**1.94**	0.29	0.62
Physics	Zero-Shot	2.25	0.57	1.54
	One-Shot	-	-	-
	Zero-Shot CoT	3.71	0.94	2.29
	Gen. Knowledge	3.80	**0.96**	**2.34**
	Dual-Prompt GK	**4.05**	0.71	1.79
Algorithms	Zero-Shot	2.95	2.04	2.78
	One-Shot	-	-	-
	Zero-Shot CoT	4.66	**2.38**	4.05
	Gen. Knowledge	**5.51**	2.34	4.29
	Dual-Prompt GK	3.60	2.02	**4.78**

Figure 15: Models' results in our benchmarks

B Comparison of grader models

Subset	Approach & Model	GPT-4	Llama	Pearson (p-value)	Spearman (p-value)
Maths	GPT-4 Dual-Prompt GK	2.25	2.09	0.7377 (5.034e-56)	0.6696 (7.642e-43)
	GPT-4o Dual-Prompt GK	3.39	3.16	0.6805 (1.013e-44)	0.6466 (3.741e-39)
	GPT-4o Gen. Knowledge	3.24	2.61	0.7902 (2.213e-69)	0.7514 (3.452e-59)
Physics	GPT-4 Dual-Prompt GK	4.43	3.58	0.7180 (1.726e-84)	0.7043 (5.132e-80)
	GPT-4o Dual-Prompt GK	6.31	5.17	0.7025 (1.955e-79)	0.7069 (7.969e-81)
	GPT-4o Gen. Knowledge	6.20	4.92	0.7816 (1.714e-109)	0.7789 (3.084e-108)
Algorithms	GPT-4 Dual Prompt GK	6.91	5.37	0.6763 (8.536e-43)	0.6635 (1.05e-40)
	GPT-4o Dual Prompt GK	8.24	6.90	0.5789 (5.925e-29)	0.5489 (1.235e-25)
	GPT-4o Gen. Knowledge	8.51	6.98	0.5128 (4.029e-22)	0.5267 (1.86e-23)

Figure 16: Comparison of GPT-4 and Llama3 as graders.

C Estimated cost of experiments

To put the experiments into perspective and to provide insight and transparency into its resource requirements, we outline the number of tokens, as well as their final cost when using the Azure OpenAI Endpoints.

In terms of tokens, the rubrics themselves contain 1.32 million tokens in total whereas the final grading contains 0.97 million tokens.

Assuming the cost of GPT-4 as per the pricing of Azure OpenAI Endpoints[4] to be 30 USD per 1M input tokens and 60 USD per 1M output tokens, the full cost of the running the grading experiments described in this paper is on the order of 200 USD.

For our benchmarks, all problem statements contain around 300k tokens, which will produce around 350k tokens of solutions. These solutions must later be graded. Such grading contains another 280k tokens. Assuming the cost of GPT-4, running one experiment on the whole dataset is in the order of 60 USD.

D Prompts and example outputs

We attach prompts used during our grading experiments below.

```
‘‘‘
(solution)
‘‘‘

From the provided sample solution, create a rubric to evaluate solutions. A maximum of 10 points can be awarded for the solution. Make sure your rubric contains all details that are needed to determine correctness of a student's solution, including all relevant equations or numeric results.
```

Figure 17: Prompt template used for rubric generation

[4] https://azure.microsoft.com/en-us/pricing/details/cognitive-services/openai-service/

```
# Grading rubric:
‘‘‘
(rubric)
‘‘‘

# Student's solution:
‘‘‘
(solution)
‘‘‘

Based on the above grading rubric, score this solution with a maximum of 10 points. Feel free to assign partial points. Briefly justify your decision, indicating for each point in the rubric whether or not the student has met it. End your answer with the text POINTS: and the number of points awarded, even if zero.
```

Figure 18: Prompt template used for grading

1. 1 point: For correctly defining the variables M, F, and P to represent the number of KMS, FKS, and KSP t-shirts respectively.
2. 1 point: For correctly defining the variables M_i, F_i, and P_i to represent the number of KMS, FKS, and KSP t-shirts in the i-th box respectively.
3. 1 point: For correctly stating that $F_i = P - P_i$ for any i.
4. 1 point: For correctly summing the equations $F_i = P - P_i$ for $i = 1, \ldots, 6$ to get $F = 5P$.
5. 1 point: For correctly stating that $M_i = F - F_i$ for any i.
6. 1 point: For correctly summing the equations $M_i = F - F_i$ for $i = 1, \ldots, 6$ to get $M = 5F$.
7. 1 point: For correctly substituting $F = 5P$ into $M = 5F$ to get $M = 25P$.
8. 1 point: For correctly stating that the total number of t-shirts is $M + F + P$.
9. 1 point: For correctly substituting $M = 25P$, $F = 5P$, and $P = P$ into $M + F + P$ to get $31P$.
10. 1 point: For correctly concluding that the total number of t-shirts is a multiple of 31.

Figure 19: Example grading rubric generated by GPT-4

The student's solution is correct and well explained. They correctly identify the goal of the problem and the approach to be taken. They correctly use the properties of the circle to determine the lengths of the sides. They correctly set up and solve the system of equations to find the angles. They correctly conclude that the triangle is equilateral. The solution is clear and correct. However, they do not draw the triangle $LO\check{D}$ and divide the sides into thirds, nor do they identify the similar triangles in the problem. They also do not apply the power of a point theorem in the advanced solution.

1. Correctly identifies the goal of the problem and the approach to be taken (1 point)
2. Correctly draws the triangle $LO\check{D}$ and divides the sides into thirds (0 point)
3. Correctly identifies the similar triangles in the problem (0 point)
4. Correctly uses the properties of similar triangles to determine the angles and sides (1 point)
5. Correctly uses the properties of the circle to determine the lengths of the sides (1 point)
6. Correctly sets up and solves the system of equations to find the angles (1 point)
7. Correctly concludes that the triangle is equilateral (1 point)
8. Correctly applies the power of a point theorem in the advanced solution (0 points)
9. Correctly concludes that all sides of the triangle are equal in the advanced solution (1 point)
10. Overall clarity and correctness of the solution (1 point)

POINTS: 7

Figure 20: Example solution grading generated by GPT-4

‘‘‘
(problem statement)
‘‘‘

Solve the given problem. It should be clear from your answer how you arrived at your result. If the problem has a numerical solution, end your answer with ANS: and the numerical value without any units. If the problem does not have a numerical solution, do not output ANS: at all.

Figure 21: Prompt template used for zero-shot experiments. Zero-shot CoT extends this by adding "Let's think step by step." at the end.

Problem:
‘‘‘
(one-shot example problem statement)
‘‘‘

Solution:
‘‘‘
(one-shot example problem solution)
‘‘‘

Problem:
‘‘‘
(problem statement)
‘‘‘

Solution:

Figure 22: Prompt template used for one-shot experiments

‘‘‘
(problem statement)
‘‘‘

Start by describing all concepts and ideas related to the problem. Then, solve the given problem. It should be clear from your answer how you arrived at your result. If the problem has a numerical solution, end your answer with ANS: and the numerical value without any units. If the problem does not have a numerical solution, do not output ANS: at all.

Figure 23: Prompt template used for generated knowledge experiments

‘‘‘
(problem statement)
‘‘‘

Describe all concepts and ideas required to solve this problem.

– (next prompt) –

Solve the given problem. It should be clear from your answer how you arrived at your result.

Figure 24: Prompt template used for dual-prompt generated knowledge experiments