SlideGuard: AI-Driven Evaluation of Graduate Student Presentation Materials

Nikolay Butakov and Maria Khodorchenko and Nikita Mazein
Daniil Gareev and **Yuri Falevskiy** and **Georgy Konev** and **Denis Nasonov**
ITMO University
Industrial AI Research Lab
Saint-Petersburg, Russia

Abstract

Preparing graduate students for effective professional communication remains a central goal of higher education, yet consistently assessing the quality of presentation slide decks - particularly in fast-growing AI/ML programs - poses significant challenges. We introduce SlideGuard, an evaluation agent that assesses slide decks against a comprehensive framework of expert-defined criteria using a visual language model. The criteria, developed in collaboration with domain experts, span visual design, narrative coherence, and argumentative structure. SlideGuard delivers explicit, interpretable justifications for its scoring decisions, and its content-hash-based caching enables efficient re-evaluation after incremental edits, reducing the time educators spend on slide deck evaluation and accelerating feedback delivery to students. We evaluate the approach on a dataset of 150 annotated slide decks and show that it detects the majority of expert-identified issues, with stronger results on structural and visual criteria and known limitations on subjective dimensions such as research quality. SlideGuard is released under the Apache 2.0 license and is available on GitHub,[1] including all criterion prompts, configuration files, and evaluation scripts to facilitate replication.

1 Introduction

Effective communication is a core objective of graduate education in AI and machine learning (ML). Presenting complex research through slide decks is a fundamental professional skill, essential for articulating ideas and defending them during project and thesis defenses. In this work, we focus specifically on AI/ML master's programs, where slide deck evaluation is a recurrent bottleneck: instructors spend substantial time on presentation-design feedback that competes with domain-level mentoring.

Interactivity is a crucial component of modern learning (Beauchamp and Kennewell, 2010), and rapid feedback loops accelerate knowledge acquisition while highlighting individual weaknesses (Hagos et al., 2022; Kochmar et al., 2020). However, as student numbers grow, providing each student with personalized guidance becomes increasingly difficult.

Large language models (LLMs) offer a path forward through the "LLM-as-a-judge" paradigm, which can deliver rapid, iterative feedback. Reliability in this setting depends on decomposing evaluation into specific, measurable criteria rather than soliciting generic comments (Kim et al., 2024). Recent multimodal LLMs (Bai et al., 2023; Chen et al., 2024b) further enable high-level reasoning over both visual and textual content. Yet most AI work on slide decks focuses on *generation* (Wang et al., 2024, 2023) rather than evaluation, leaving a gap in providing critical feedback on rhetorical effectiveness, design quality, and scientific argumentation.

To address this gap, we introduce SlideGuard, an evaluation agent for rigorous, criteria-driven slide deck assessment. Unlike prior systems that focus on slide *generation* or structural extraction, SlideGuard is the first open-source tool that provides holistic, criteria-based *evaluation* combining visual design analysis, narrative coherence checking, and argumentative quality assessment. Our key contributions are:

1. A multi-level taxonomy of common slide deck issues - with concrete, operationalized criteria at both the slide level (e.g., checking whether title text matches slide content, detecting unexplained abbreviations, verifying required elements on title slides) and the deck level (e.g., evaluating section dependencies in sto-

[1] https://github.com/Industrial-AI-Research-Lab/SlideGuard

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), pages 785–793

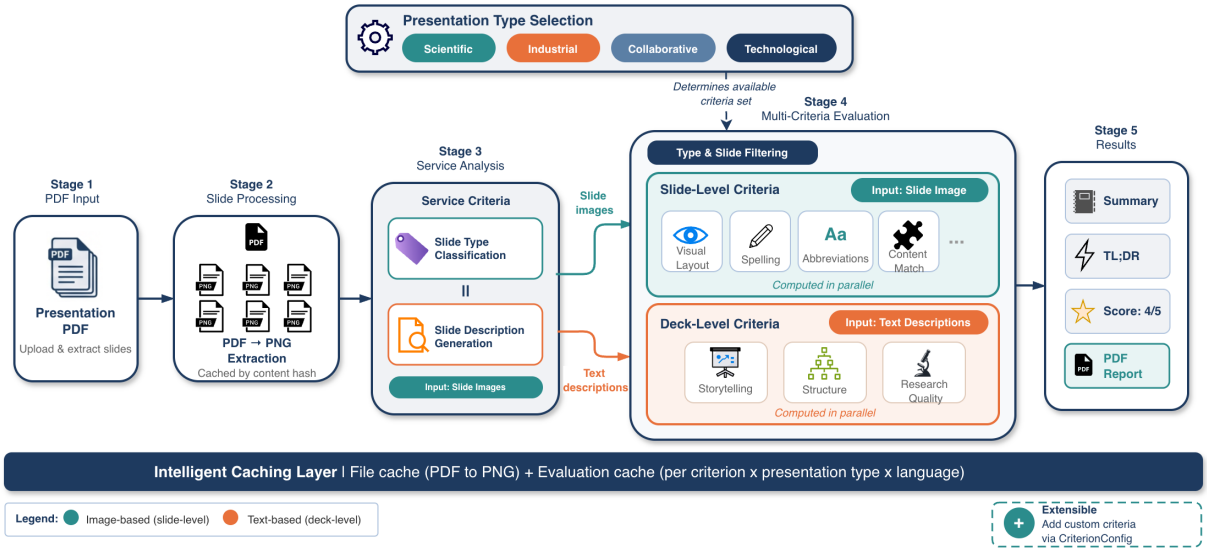

Figure 1: The workflow of the AI agent for checking slide decks issues

rytelling, checking structural completeness) - grounded in expert interviews and domain-specialist annotations.

2. A graph-based evaluation pipeline that leverages multimodal LLMs with contextual prompting and structured-output chain-of-thought reasoning to detect issues and deliver actionable feedback.

3. A comprehensive experimental study demonstrating the effectiveness of the proposed approach on both automated metrics and human evaluation.

2 Problem Analysis and Criteria Design

2.1 Challenges in Slide Deck Evaluation

A common approach to teaching presentation skills relies on templates, guidelines, and practice sessions. In practice, however, a significant amount of time is spent re-explaining basics of slide construction and information structuring, even when preparation materials are provided. This leaves less time for domain-level mentoring.

To quantify this, we interviewed five associate professors in AI/ML (5+ years of teaching experience) and analyzed video recordings of thesis presentations from master's programs in artificial intelligence. For each of three consecutive semesters, we analyzed 25 presentations. Figure 2 shows a general downward trend in the total number of comments, indicating improvement over time. However, issues persist across semesters, motivating the need for automated evaluation to reduce educator workload and provide earlier feedback.

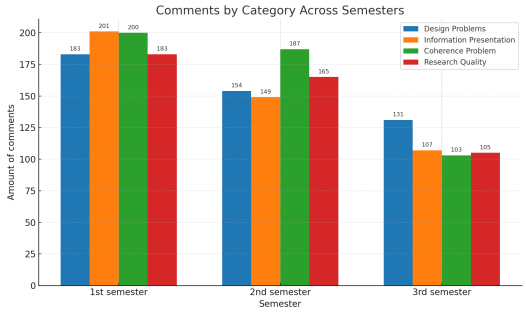

Figure 2: Persistence of different issue types across semesters.

2.2 Taxonomy and Criteria

From expert feedback and presentation analysis, we developed a taxonomy of common issues at two complementary levels (Figure 3). Each criterion specifies concrete error patterns and evaluation rules rather than generic quality dimensions:

- **Slide-level:** visual design (arrangement, color/fonts), information presentation (abbreviations, link availability, title quality, orthography, title–content match, graphic–content match), and type-specific content (track justification, novelty, related works, key results, industrial applicability).

- **Deck-level:** coherence (storytelling, structure completeness) and research quality.

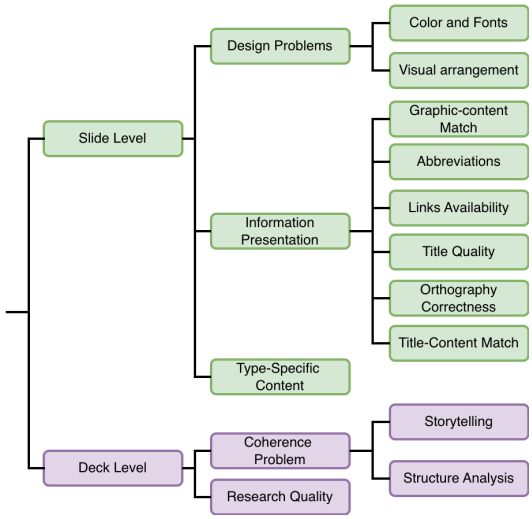

Figure 3: The proposed taxonomy of presentation issues.

This two-level decomposition separates checks requiring per-slide visual evidence from those requiring a holistic view of the full deck. Our taxonomy was developed with and validated by AI/ML supervisors; criteria such as visual arrangement, abbreviations, spelling, and storytelling address presentation skills common across STEM disciplines, while others (e.g., research quality, track justification) are specific to thesis defense formats. The framework is designed to be extensible: domain-specific criteria can be added without modifying the evaluation pipeline (Appendix B), enabling adaptation to other fields where the taxonomy would need different content-level criteria.

To build the taxonomy, we collected 150 real student submissions from master's programs in AI/ML across four consecutive semesters (Fall 2024–Spring 2025), averaging 12 slides per deck. Each presentation was evaluated by at least 3 of 5 associate professors, who provided free-form textual feedback. We found no contradictions between evaluators, though they offered complementary perspectives. Their unified comments constitute the annotated dataset used for evaluation.

Each criterion is operationalized as a structured prompt with six components: (1) a role and task description, (2) key elements to evaluate, (3) common problems to identify (with specific examples, e.g., "text too small for easy reading" or "chart axis labels that do not correspond to the surrounding text"), (4) an explicit *not-a-problem* list to suppress false positives (e.g., "numbered lists instead of bullet points"), (5) evaluation guidelines that provide chain-of-thought scaffolding with a step-by-step analysis plan, and (6) a JSON response schema requiring each detected issue to carry a severity level (0–3), the specific problematic element, and an actionable suggestion. For instance, the *title-content match* criterion instructs the model to (i) extract the title, (ii) independently formulate the slide's main idea, (iii) compare them, and (iv) suggest a revised title if they diverge. The *storytelling* criterion describes inter-section dependencies via a predefined graph (Figure 4). All prompts are released as part of the open-source repository.

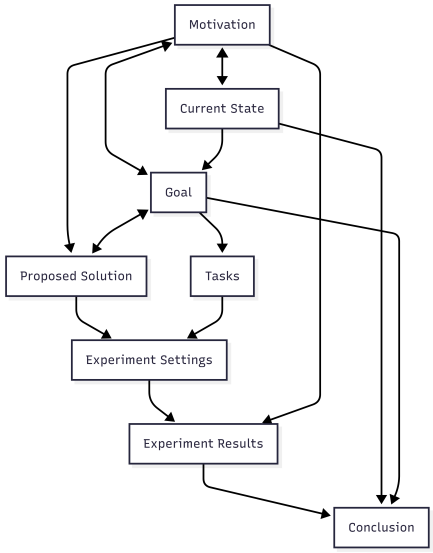

Figure 4: Dependency graph used by the storytelling criterion.

2.3 Formal Problem Statement

Let $D = \{s_1, \ldots, s_n\}$ be a presentation deck of n slides. We define two disjoint sets of evaluation criteria: m slide-level criteria $\mathcal{C}^{slide} = \{c_1^{slide}, \ldots, c_m^{slide}\}$ that operate on individual slides, and p deck-level criteria $\mathcal{C}^{deck} = \{c_1^{deck}, \ldots, c_p^{deck}\}$ that require a holistic view of the full deck.

For each slide-level criterion c_i^{slide}, let f_i^{slide} denote the corresponding LLM-based evaluation function. It takes as input a slide image x_j (the raster rendering of slide s_j) together with a criterion-specific evaluation context C_i (the structured prompt, output schema, and chain-of-thought scaffolding), and returns a scalar score $e_{ij} \in [1, 5]$:

$$e_{ij} = f_i^{slide}(x_j, C_i), \quad \bar{e}_i = \frac{1}{|\mathcal{A}_i|} \sum_{j \in \mathcal{A}_i} e_{ij} \qquad (1)$$

where $\mathcal{A}_i \subseteq \{1, \ldots, n\}$ is the subset of slides to which criterion c_i is applicable (determined by slide

type and infographics constraints), and $\bar{e}_i$ is the averaged score across applicable slides.

For each deck-level criterion c_k^{deck}, let g_k^{deck} denote its evaluation function, which operates on a textual representation of the full deck. This representation is constructed by first applying an LLM description generator $\mathcal{G}$ to each slide image, then concatenating ($\oplus$) the resulting descriptions:

$$e_k^{deck} = g_k^{deck}(D^{desc}, C_k), \quad D^{desc} = \bigoplus_{j=1}^{n} \mathcal{G}(x_j) \tag{2}$$

where D^{desc} is the aggregated textual deck description and C_k is the criterion-specific context. Each $e_k^{deck} \in [1, 5]$.

The overall presentation score is a weighted combination:

$$\mathcal{S}(D) = \sum_{i=1}^{m} w_i \, \bar{e}_i + \sum_{k=1}^{p} w_k \, e_k^{deck} \tag{3}$$

where $w_i, w_k \geq 0$ are criterion weights satisfying $\sum_i w_i + \sum_k w_k = 1$. In the current implementation, all criteria are weighted equally.

3 System Description

SlideGuard is an evaluation agent that assesses PDF presentations against a configurable set of quality criteria and produces per-slide and per-deck feedback, an aggregated score, a natural-language summary, and an exportable PDF report. The system is built on LangGraph for workflow orchestration and LangChain for LLM interaction, and is compatible with both OpenAI API endpoints and self-hosted inference servers (e.g., vLLM).

3.1 Evaluation Pipeline

The pipeline is implemented as a directed acyclic graph (DAG) assembled dynamically based on the selected criteria (Figure 1). It comprises three sequentially dependent phases.

Phase 1: Service criteria. The input PDF is converted to per-slide raster images encoded in base64 for use with OpenAI-compatible APIs. Two service criteria then execute in parallel: a *slide type classifier* assigns each slide semantic labels from a predefined taxonomy (e.g., "Title slide", "Problem Statement", "Experimental Results"), and a *description generator* produces a structured textual summary of each slide's content and visual elements. Both operate on slide images via multimodal LLM prompts and are prerequisite to all subsequent steps.

Phase 2: Parallel evaluation. Two concurrent subgraphs execute the main evaluation. The *slide-level subgraph* evaluates each criterion in parallel across applicable slides. An applicability filter restricts each criterion to eligible slides based on classified slide types, the declared presentation type, and optional constraints (e.g., presence of infographic content). Each criterion queries the LLM with the slide image and parses the response into a structured output via a retry-capable parser. The *deck-level subgraph* operates concurrently: it aggregates per-slide descriptions into a textual deck representation and evaluates criteria such as storytelling, structure completeness, and research quality against this text input.

An important consequence of this design is that SlideGuard does *not* require all talking points to be written on the slides. Slide-level criteria operate directly on the slide *image*, assessing visual properties (layout, fonts, chart legibility) rather than demanding exhaustive text. Deck-level criteria operate on LLM-generated *descriptions* of what is visually depicted, not on verbatim slide text; the description generator infers content from headings, diagrams, and spatial arrangement. This avoids the failure mode of essay-style evaluation that would penalize concise, well-designed slides.

This two-track design also offers architectural advantages: (i) each LLM call focuses on a narrow scope, improving detection quality; (ii) slide-level and deck-level evaluations run in parallel, improving throughput; (iii) decks of arbitrary length are supported, since deck-level criteria operate on compressed text descriptions rather than raw images; and (iv) per-slide caching enables efficient re-evaluation after incremental edits.

Phase 3: Summarization. After both evaluation subgraphs complete, a summary processor filters and prioritizes findings using adaptive severity thresholds, then prompts the LLM to generate a long-form summary and a condensed TL;DR. An overall score is computed as the weighted mean of all criterion scores.

3.2 Criteria Framework

The criteria system follows a registry pattern that separates criterion definition from evaluation logic. Each criterion is specified declaratively with: a unique identifier, a target level (slide or deck), prompt templates, an output schema specification, applicability constraints, and optional post-

processor functions.

Applicability filtering. Each criterion can declare whitelists or blacklists of slide types and presentation types, and a flag indicating whether infographic content is required. These constraints are evaluated at runtime against the service criteria outputs, ensuring that, for example, title slide quality assessment runs only on title slides.

Structured output. Output schemas are generated programmatically: most criteria produce a list of scored items (each with severity, detected element, and actionable suggestion) plus an overall score. This ensures structurally consistent, machine-readable output across all criteria.

Post-processing. Composable post-processor functions refine LLM outputs after parsing, including severity-based filtering, domain-specific whitelisting (e.g., suppressing known abbreviations), and deduplication.

Extensibility. Adding a new criterion requires only defining its prompt and appending a configuration entry (Appendix B). The pipeline, caching layer, and user interface automatically incorporate it without further code changes. To be more specific, for adaptation to new domains, we separate criteria into two groups. Domain-invariant criteria, such as visual arrangement, font readability, spelling, title–content match, and storytelling, can be reused with minimal changes. Domain-specific criteria, such as research quality, industrial applicability, or track justification, require a short calibration stage. In practice, this involves collecting a small set of representative decks, asking instructors to list recurring problems, converting these problems into criterion-specific positive and negative examples, and expanding the not-a-problem list to avoid penalizing acceptable domain conventions. This workflow reduces prompt development from designing a full rubric from scratch to calibrating a small number of domain-specific checks.

3.3 User Interface

SlideGuard is designed for graduate students preparing thesis or project defenses and for instructors who supervise them. It provides a Gradio-based web interface (Appendix A Figure 5) in which users upload a PDF, select a presentation type (which filters available criteria), choose evaluation criteria, and inspect per-slide feedback with an interactive slide viewer. Results can be exported as a PDF report. The system supports multilanguage output (English/Russian), role-based access control, and user management for classroom deployments.

Deployment and data handling. SlideGuard is designed for on-premises deployment so that student data remains within institutional infrastructure. Uploaded PDFs are converted to per-slide images and stored in a local content-addressed cache alongside evaluation results; temporary files created during a UI session are cleaned up on session end. When a locally hosted model (e.g., vLLM) is used as the inference backend, no student data leaves the institution's network. The caching layer is keyed by a content hash of each slide, so re-uploading a revised deck triggers re-evaluation only for slides whose content has changed, rather than reprocessing the entire presentation.

4 Experiments and Results

We evaluated SlideGuard on a held-out subset of 30 slide decks (approximately 360 slides) not used during taxonomy development. We tested two LLMs: GPT-4o and a locally deployed Qwen2.5-VL-72B-Int4 (Bai et al., 2025). Experiments with the local model were conducted on two NVIDIA H100 GPUs (80 GB each).

As the evaluation metric, we measure *detection rate*: the percentage of expert-annotated issue comments for which a matching automatically generated comment exists. This metric captures recall (how many real issues the system finds) but does not measure precision (how many of the system's findings are genuine issues); a precision study is left for future work. Results are presented in Table 1. Here, human evaluation refers to expert-annotated presentation issues used as the reference set for detection, rather than to a deployment user study with students.

The evaluation agent achieves 73.7% and 80.7% average detection rates with Qwen-VL and GPT-4o, respectively. Performance is notably strong on structure completeness and storytelling - among the most important criteria, as they directly affect how effectively a presentation communicates its core message. Abbreviation detection and spelling also reach near-perfect scores.

We include Qwen2.5-VL-72B-Int4 to test whether the system can operate under an on-premises deployment scenario, which is important

Criterion	Level	Qwen-VL	GPT-4o
Structure completeness	deck	100.0	100.0
Storytelling	deck	78.3	83.4
Research quality	deck	38.5	45.0
Color and fonts	slide	75.0	100.0
Fact link availability	slide	75.0	75.0
Graphic–content match	slide	47.8	61.5
Title–content match	slide	66.7	72.5
Title slide quality	slide	77.8	79.8
Visual arrangement	slide	64.5	70.5
Abbreviations	slide	87.2	100.0
Spelling	slide	100.0	100.0
Average		**73.7**	**80.7**

Table 1: Percentage of expert-annotated issues detected by the evaluation agent (%).

for educational settings where student materials may be sensitive or resources may be limited. The prompts were kept identical across GPT-4o and Qwen-VL to test criterion portability rather than model-specific prompt tuning. Consequently, the reported local-model results should be interpreted as a conservative estimate: additional calibration or model-specific prompt optimization may improve local deployment performance, but was not used here to preserve comparability.

The weakest results are on research quality (38.5-45.0%) and graphic-content match (47.8–61.5%). The research-quality criterion should therefore be interpreted as a triage signal rather than as an automatic assessment of scientific merit. In the current system, this criterion checks whether the deck provides visible evidence for standard thesis-defense expectations: motivation, novelty claim, methodological consistency, experimental support, and connection between results and conclusions. It is not intended to replace advisor judgement about the actual novelty or correctness of the research. This distinction is important for deployment: SlideGuard can flag missing or weakly presented evidence, but domain experts remain responsible for evaluating the underlying scientific contribution. Graphic-content matching may be improved through few-shot prompting with aligned and misaligned examples.

Although full precision annotation requires a separate expert study, SlideGuard includes several design choices intended to reduce false positives in deployment. First, each criterion contains an explicit not-a-problem list, which prevents the model from penalizing acceptable presentation choices such as numbered lists or concise slide text. Second, post-processing suppresses low-severity or duplicated findings and can whitelist domain-specific abbreviations. Third, the interface presents generated comments as inspectable recommendations rather than final grading decisions, allowing instructors or students to accept, revise, or ignore individual suggestions. Thus, recall-oriented evaluation reflects whether the system can surface expert-relevant problems, while the deployed workflow mitigates the risk of unsupported automatic penalties.

Since we are not aware of a published system that performs holistic, criteria-based slide deck evaluation, comparison with prior tools is not directly available. However, a natural baseline is a single whole-deck VLM prompt that evaluates the presentation without slide-type filtering, per-criterion decomposition, or reusable per-slide caching. We use expert annotations as the primary reference because the goal of SlideGuard is not only to assign a global quality score, but to recover localizable, actionable issues that can be mapped to concrete slides and criteria. The decomposition into service criteria, slide-level checks, and deck-level checks is therefore motivated by three requirements that a whole-deck prompt does not naturally satisfy: applicability control, interpretable localization of feedback, and efficient re-evaluation of edited slides.

These results suggest that SlideGuard can serve as a first-line evaluation tool: students can fix foundational issues independently, leaving more time for in-depth discussion of scientific and professional questions with their advisors. An example of the agent's output on the storytelling criterion is shown in Appendix A.

5 Related Works

LLM-as-a-Judge approach is a commonly used technique not only for evaluating the outputs of other LLMs (Zheng et al., 2023; Ke et al., 2024; Ling et al., 2023) but also for assessing human responses, illustrations, and interactive tasks (Chen et al., 2024a; Song et al., 2024). Results of evaluations are proven to be close with human judgments opening opportunities for large-scale and cost-efficient feedback systems in education.

Within the educational domain, LLMs and MLLMs have been applied to diverse tasks such as automated essay scoring (Song et al., 2024; Feng et al., 2024), assessment of student explanations

(Carpenter et al., 2024), and evaluation of self-explanations in programming contexts (Chapagain and Rus, 2025). These studies provide a valuable justification on AI-based system usage to complement or partially replace human graders, while still ensuring reliability and fairness.

Prior AI work on presentations focuses predominantly on generation, not evaluation. OutlineSpark (Wang et al., 2024) generates slides from computational notebooks, while Slide4N (Wang et al., 2023) uses human-AI collaboration for content structuring. Critical thinking support systems (Inoue et al., 2023) recommend slides or generate questions but omit design/rhetorical analysis, and accessibility tools such as Slide Gestalt (Peng et al., 2023) automatically extract structural information from slides. While these methods support presentation preparation, they do not provide holistic, criteria-based evaluation of visual design, rhetorical effectiveness, or argument flow.

6 Conclusion and Future Work

We introduced SlideGuard, a framework for systematically evaluating student slide decks using multimodal large language models. By analyzing expert interviews and student presentations from AI/ML master's programs, we identified persistent challenges students face, developed a multi-level taxonomy of operationalized criteria, and curated an annotated dataset of 150 slide decks capturing both slide-level design flaws and deck-level coherence problems. Our evaluation agent combines contextual prompting, chain-of-thought reasoning, and multimodal analysis to detect issues and provide structured feedback. While the system shows strong results on structural and visual criteria, performance on subjective dimensions such as research quality remains limited, and our evaluation is constrained to a single institution and discipline. Nevertheless, the results suggest that SlideGuard can serve as a useful first-line feedback tool, enabling students to iterate independently on foundational issues and freeing instructors for domain-level mentoring.

In future work, we plan to extend SlideGuard to evaluate presentation speech quality and assess answers to synthesized questions, creating a full presentation simulator for better preparation and stress reduction. We also aim to increase interactivity through a dialogue system that allows students to discuss slide deck improvement techniques with the agent. Future work will also evaluate SlideGuard in a classroom deployment by measuring how feedback affects student revisions and instructor workload, and will explore integration into presentation-authoring environments, such as an Office plugin, so that students can receive feedback directly during slide preparation.

Limitations

Our evaluation is limited in several respects. First, the held-out test set of 30 decks is small; while sufficient for criterion-level trends, it limits the statistical power of fine-grained comparisons. Second, we measure detection coverage (whether expert-annotated issues are found) but not precision (whether the system's additional findings are valid), which requires a separate annotation study. Third, the current taxonomy and dataset are drawn from AI/ML master's programs at a single institution; generalization to other disciplines and educational contexts remains to be validated. Fourth, research quality evaluation remains the weakest criterion (38.5–45.0%), likely because it requires deeper domain knowledge than can be conveyed through prompting alone; this limits the system's utility for assessing scientific rigor.

Acknowledgments

This work supported by the Ministry of Economic Development of the Russian Federation (IGK 000000C313925P4C0002), agreement No139-15-2025-010

References

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *Preprint*, arXiv:2308.12966.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Gary Beauchamp and Steve Kennewell. 2010. Interactivity in the classroom and its impact on learning. *Computers Education*, 54(3):759–766. Learning in Digital Worlds: Selected Contributions from the CAL 09 Conference.

Dan Carpenter, Wookhee Min, Seung Lee, Gamze Ozogul, Xiaoying Zheng, and James Lester. 2024. Assessing student explanations with large language models using fine-tuning and few-shot learning. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 403–413, Mexico City, Mexico. Association for Computational Linguistics.

Jeevan Chapagain and Vasile Rus. 2025. Automated assessment of student self-explanation in code comprehension using pre-trained language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(28):28996–29003.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024a. Mllm-as-a-judge: assessing multimodal llm-as-a-judge with vision-language benchmark. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Haiyue Feng, Sixuan Du, Gaoxia Zhu, Yan Zou, Poh Boon Phua, Yuhong Feng, Haoming Zhong, Zhiqi Shen, and Siyuan Liu. 2024. Leveraging large language models for automated chinese essay scoring. In *Artificial Intelligence in Education*, pages 454–467, Cham. Springer Nature Switzerland.

Misgina Tsighe Hagos, Kathleen M. Curran, and Brian Mac Namee. 2022. Impact of feedback type on explanatory interactive learning. In *Foundations of Intelligent Systems*, pages 127–137, Cham. Springer International Publishing.

Saki Inoue, Yuanyuan Wang, Yukiko Kawai, and Kazutoshi Sumiya. 2023. Encouraging critical thinking support system: Question generation and lecture slide recommendations. In *Proceedings of the Tenth ACM Conference on Learning @ Scale*, L@S '23, page 287–291, New York, NY, USA. Association for Computing Machinery.

Pei Ke, Bosi Wen, Andrew Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, Jie Tang, and Minlie Huang. 2024. CritiqueLLM: Towards an informative critique generation model for evaluation of large language model generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13034–13054, Bangkok, Thailand. Association for Computational Linguistics.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024. Prometheus: Inducing fine-grained evaluation capability in language models. *Preprint*, arXiv:2310.08491.

Ekaterina Kochmar, Dung Do Vu, Robert Belfer, Varun Gupta, Iulian Vlad Serban, and Joelle Pineau. 2020. Automated personalized feedback improves learning gains in an intelligent tutoring system. In *Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, July 6–10, 2020, Proceedings, Part II*, volume 12164 of *Lecture Notes in Computer Science*, pages 140–146, Cham. Springer.

Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. 2023. Deductive verification of chain-of-thought reasoning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Yi-Hao Peng, Peggy Chi, Anjuli Kannan, Meredith Ringel Morris, and Irfan Essa. 2023. Slide gestalt: Automatic structure extraction in slide decks for non-visual access. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Yishen Song, Qianta Zhu, Huaibo Wang, and Qinhua Zheng. 2024. Automated essay scoring and revising based on open-source large language models. *IEEE Transactions on Learning Technologies*, 17:1880–1890.

Fengjie Wang, Yanna Lin, Leni Yang, Haotian Li, Mingyang Gu, Min Zhu, and Huamin Qu. 2024. Outlinespark: Igniting ai-powered presentation slides creation from computational notebooks through outlines. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

Fengjie Wang, Xuye Liu, Oujing Liu, Ali Neshati, Tengfei Ma, Min Zhu, and Jian Zhao. 2023. Slide4n: Creating presentation slides from computational notebooks with human-ai collaboration. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

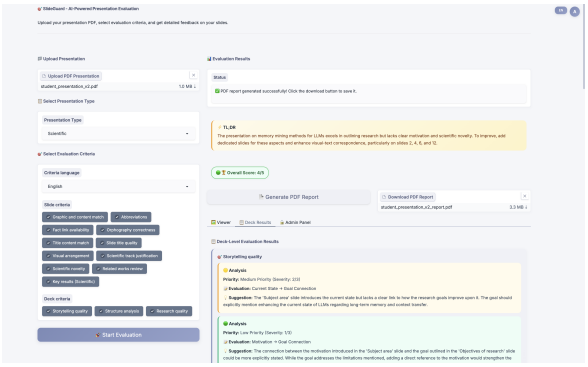

Figure 5: Screenshot of the SlideGuard web interface.


```
"deck_evaluations": {
    "evaluations": {
        "deck_storytelling": {
            "evaluation_results": [
                {
                    "severity": 3,
                    "evaluation_element": "Goal → Tasks Connection",
                    "evaluation_suggestion": "The tasks listed in the
                        'Goal and objectives' slide do not clearly
                        specify how they directly support achieving the
                        goal. For instance, the task 'Develop data
                        augmentation pipeline for dataset enrichment'
                        lacks detail on how it contributes to the overall
                        goal of developing a few-shot method for intent
                        classification. To improve this, consider adding
                        more specific details on how each task supports
                        the goal."
                },
                {
                    "severity": 2,
                    "evaluation_element": "Solution → Experiments
                        Connection",
                    "evaluation_suggestion": "The connection between the
                        proposed solutions and the experiment settings
                        could be clearer. While the slides cover various
                        aspects of the proposed solution, the transition
                        to the experiment settings does not explicitly
                        show how the experiments test the proposed
                        solution. Adding a slide that bridges the gap
                        between the proposed solution and the experiment
                        settings would help clarify this connection."
                },
```


Figure 6: Example output of the evaluation agent on the storytelling criterion.

A SlideGuard Interface and Output Example

B Defining a New Criterion

Adding a criterion to SlideGuard requires two steps: writing a prompt and registering a configuration entry. Below is a minimal example of a slide-level criterion that checks whether data visualizations include axis labels.

Step 1: Define the prompt.

```
prompt = """
You are an expert in data visualization.
Evaluate whether charts and graphs on the
slide have clearly labeled axes.

## Key Elements to Evaluate:
- Axis titles present and descriptive
- Units of measurement specified
- Legend present when multiple series shown

## Evaluation Guidelines:
Provide your reasoning in a Thought section,
then return the evaluation in the Answer
section following the format instructions.
"""
```

Step 2: Register the configuration.

```
CriterionConfig(
  id=Criteria.slide_axis_labels,
  target=CriteriaTarget.slide,
  description="Check axis labels on charts",
  agent_prompt_template=prompt,
  task_prompt_template=BASE_SLIDE_TASK_PROMPT,
  output=OutputSpec(
    kind=OutputKind.scored_list,
    item_spec=ScoredListItemSpec(
      element_description="Chart with issue",
      suggestion_description="Suggested fix",
    ),
  ),
  category="visual",
  applicability=Applicability(
    requires_infographics=True,
    exclude_slide_types=[
      SlideType.TITLE_SLIDE,
      SlideType.END_SLIDE,
    ],
  ),
)
```

The configuration declares that this criterion applies only to slides containing infographics (as detected by the service phase) and excludes title and end slides. The output schema - a scored list of items with severity, element, and suggestion fields - is generated automatically. Once appended to the configuration list, the new criterion is available in the CLI, web UI, and Python API without any further code changes.