SLIDEGATE: BENCHMARKING PRESENTATION GENERATION WITH DIMENSION-SPECIFIC VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Automated presentation generation has entered practical use, enabling users to obtain complete slide decks from natural-language requests and optional reference materials. Such requests often involve delivery requirements, including slide count, formatting, source grounding, and mandatory content, while also requiring slides to be free of severe visual defects such as occlusion and cropping. We introduce SlideGate, a benchmark comprising 40 presentation-generation tasks across seven application scenarios and five input forms, with 2,191 task-specific verification items and a protocol that separately reports instruction following, content quality, and visual presentation. Applying SlideGate to 240 outputs from six publicly available commercial slide-generation systems, we obtain binary content-item judgments that agree with two independent blinded annotators at rates of 80.7% and 85.6%, respectively, and system rankings based on mean per-slide visual scores that correlate with human preference rankings at Spearman $\rho = 0.829$. Across these outputs, SlideGate detects 124 decks with blocking failures, including hard instruction violations, missing required content, or page-level cropping or occlusion. Using scoring schemes based on the published criteria of six representative evaluation frameworks, the initial grouping analysis finds that three schemes assign significantly lower scores to decks with detected blocking failures, but the score gaps are small. When thresholds are set to detect approximately 90% of failing decks in leave-one-task-out validation, these three schemes incur false-positive rates of 76.8% to 98.8%. These findings highlight the value of combining task-specific verification with explicit failure diagnosis to reveal delivery issues that holistic scores may obscure.

1 INTRODUCTION

Presentations communicate information to a particular audience in a particular setting. Requests for presentation generation not only specify a topic but often also involve requirements suited to the intended setting and audience, including slide count, formatting, mandatory content, and source grounding; generated slides must also avoid severe visual defects such as cropping and occlusion. A central question in evaluating these systems is whether their outputs satisfy these requirements. We refer to hard instruction failures, missing required content, and page-level cropping or occlusion collectively as *blocking failures*, and focus on their joint diagnosis.

Existing presentation evaluations use reference similarity, content and design ratings, or specialized defect criteria. These approaches capture useful aspects of quality, but aggregate scores leave unclear which requested requirements are unmet and which information on a rendered page is damaged. We collect 240 outputs from six publicly available commercial systems and evaluate them with scoring schemes based on six representative frameworks (Section 5.2). SlideGate detects 124 decks with at least one blocking failure: 33 have hard instruction failures, 71 have missing required content, and 60 contain cropping or occlusion; the categories overlap. When outputs are grouped by whether a blocking failure was detected, three of the six scoring schemes assign lower scores to the failure group (Mann–Whitney U), yet the gaps are small, and a threshold that catches about 90% of failing decks also flags most decks without a detected failure.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Instruction failures include omitted mandatory sections and violated explicit data rules, as well as simpler count violations: in an academic-talk task requiring 5–10 slides, Manus produces 12. In a company-introduction task, a Manus deck that fails a hard instruction still ranks first among the same task's outputs under the SlideAudit, PPTBench, and AutoPresent scores. Content failures can arise similarly: a training task explicitly requires the second lesson plan as its source, but an output uses the first lesson. Visual failures can affect the information itself: in a physics deck, cropped teaching content and obstructed formulas recur across several pages. These errors require checking constraints, consulting materials, or inspecting rendered pages; their consequences are not described by a favorable quality ranking alone.

We introduce SlideGate, a presentation-generation benchmark comprising 40 tasks and outputs from six systems—AnyGen, Kimi, Doubao, Skywork, MiniMax, and Manus. Tasks span document attachments, editing instructions, template references, and other input forms. Its evaluation protocol organizes evidence into instruction following, content quality, and visual presentation. Through independent candidate generation, critic review, and rule-based validation, SlideGate constructs task-specific instruction and content-coverage specifications from requests and parseable sources, freezes them across systems, and verifies outputs against them. Page-level visual assessment complements these task-specific checks. Together, these components provide item- and page-level failure evidence alongside quality scores. An anonymized repository with the tasks, frozen specifications, judgments, and evaluation code is at https://anonymous.4open.science/r/slidegate.

Our contributions are:

1. **Benchmark resource.** SlideGate covers 40 tasks across seven scenarios and five input forms, together with task-specific verification specifications and 240 outputs from six commercial presentation-generation systems.
2. **Evaluation protocol.** We combine request-grounded instruction checks, source-aware content assessment, and rendered-page visual diagnosis. Rule-based validation and frozen task specifications provide a common basis for comparing systems, while explicit failure records complement quality scores.
3. **Empirical findings.** We identify 124 decks with blocking failures and characterize the different weaknesses of current systems. Comparison-score analyses show limited failure-screening performance, and a same-backbone experiment reveals how generation-workflow improvements affect different capabilities. Programmatic references, blinded annotations, and designer preferences validate the corresponding judgments.

2 Related Work

Presentation generation and quality evaluation. Document-to-slide systems such as DOC2PPT and D2S couple summarization with visual composition (Fu et al., 2022; Sun et al., 2021). Recent systems use language-driven planning, editing, and code generation (Zheng et al., 2025; Ge et al., 2025; Jang et al., 2026). Their evaluations include both reference-based comparisons and reference-free quality ratings; they treat evaluation as rating the quality of a generated deck. PPTEval evaluates content, design, and coherence; its design description explicitly considers readability and overlapping elements (Zheng et al., 2025). AutoPresent's reference-free evaluation scores text, images, layout, and color, with layout criteria covering overlap and page boundaries (Ge et al., 2025). LecEval combines slides and lecture text in a learned educational-quality evaluator (Yin et al., 2025). These dimensions provide useful quality signals; whether their scores identify specific failures in complete generated decks is an empirical question.

Specialized presentation benchmarks. A second line of work makes evaluation itself the task, with dedicated data and annotations. PPTBench separates detection, understanding, modification, and generation tasks; its detection tasks include overlap and out-of-bounds elements (Huang et al., 2025). SlideAudit develops a design-defect taxonomy and annotated slide dataset for evaluating defect detection (Zhang et al., 2025). SlidesGen-Bench combines source-document question answering, computational aesthetics, and editability (Yang et al., 2026). Prior work therefore already evaluates visual quality through defect taxonomies, design ratings, and computational aesthetics, largely on its own data and labels. Our comparison applies the quality scores of six representative frameworks to the same outputs (Appendix E).

Table 1: Task composition. Scenarios and input forms are separate marginal distributions.

Scenario	Tasks	Main input form	Tasks
Training and teaching	8	Documents and requirements	17
Data analysis	8	Detailed instructions	6
Academic talks	6	Brief requests	6
Business proposals	5	Existing decks and edits	6
Project planning	5	Templates and materials	5
Work reports	5		
Promotion presentations	3	**Total**	**40**

Task-specific verification and model judges. PPTArena evaluates presentation editing, while PresentBench provides manually authored binary rubrics for slide-generation instances (Ofengenden et al., 2025; Chen et al., 2026). In language evaluation, IFEval uses verifiable instructions, FollowBench studies increasing constraints, and ComplexBench combines rules with model judgments for compositional requirements (Zhou et al., 2023; Jiang et al., 2024; Wen et al., 2024). WildBench and CheckEval further establish task-specific and decomposed evaluation (Lin et al., 2025; Lee et al., 2025). SlideGate develops this approach for heterogeneous presentation requests and materials: it constructs rule-validated frozen specifications grounded in request spans and source materials, applies programmatic references for objectively verifiable constraints such as slide count, and studies failures across commercial systems. We evaluate judgment reliability against references matched to each dimension, following the model-judge validation approach of G-Eval and MT-Bench (Liu et al., 2023; Zheng et al., 2023). LLMBar's distinction between appealing responses and instruction adherence is also relevant to interpreting quality scores (Zeng et al., 2024).

3 Benchmark

Each instance contains a natural-language request, optional materials, and metadata describing its scenario and input form. Requests range from brief topics to detailed generation and editing instructions. Materials include documents, spreadsheets, existing decks, and visual templates; attachments whose text or structural content can be extracted by the evaluation pipeline serve as parseable source materials that support content verification, while visual-only inputs such as image files provide reference context. The evaluation uses the request supplied to the generation system.

SlideGate contains 40 tasks across seven scenarios and five input forms (Table 1). Nine tasks combine multiple input types, and 28 include one or more attachments. The evaluation includes 2,191 task-specific verification records and 240 decks from six commercial systems. Appendix A provides task examples and the instance schema.

4 Evaluation Method

4.1 Constructing and Applying Verification Specifications

SlideGate first constructs instruction and content-coverage specifications for each task, then evaluates the generated decks and aggregates the resulting judgments. Three candidates are independently generated, consolidated, reviewed by a critic, and adjudicated against the request and source evidence. Rules require a supporting request quotation for every instruction item, prohibit requirements inferred from role conventions, scenario labels, or file types not stated in the request, validate criterion types and granularity, and remove duplicate obligations. Accepted items receive stable identifiers and are frozen before cross-system evaluation. **The instruction and coverage specification is identical for all systems solving the same task.**

The evaluator returns an item verdict together with evidence pages or source references. Deterministic rules resolve supported numerical constraints, and a structured aggregation procedure maps valid verdicts to scores. Visual checks operate directly on rendered pages, including for tasks with no attachments or detailed instructions. Table 2 lists the criteria; Appendix C gives the rule layer and specification procedure.

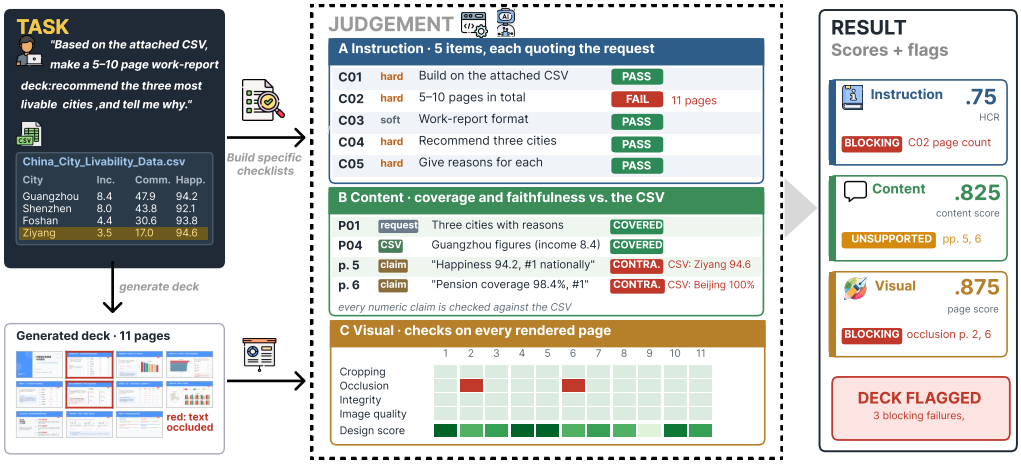


Figure 1: SlideGate on one task. Left: a work-report request and its CSV attachment are frozen into checkable items; the 11-page deck is below. Middle: instruction items are judged against request quotations (A); coverage items and numeric claims are checked against the CSV (B); every page receives defect checks and a design score (C). Right: the three dimensions stay separate scores. The page-count failure and the occluded pages are blocking flags; claims that contradict the CSV lower the faithfulness component of the content score and are also listed separately.

Table 2: Criteria and judgment units. Detailed definitions and rating anchors appear in Appendix B.

Dimension	Criteria	Judgment unit and output
Instruction	Count/format; language; layout/structure; explicit mandatory content; template retention; additions/revisions	Frozen request item: pass, partial, fail, or NA; HCR and hard failures
Content	Coverage; depth: presence (L0), specificity (L1), appropriateness (L2); source faithfulness	Content item or claim: coverage, depth, and supported/derived/unsupported judgments
Visual defects	Cropping; occlusion; content integrity; image quality	Binary checks per page; counts by defect and affected deck
Visual design	Alignment; spacing; component consistency; legibility; type hierarchy; text flow	Six page criteria, each 0/1/2
Visual aesthetics	Hierarchy/reading path; visual language; overall impression; cross-page consistency	Three page criteria, each 0–3; deck-level consistency

4.2 Instruction Following

Instruction checks concern requirements stated in the request. Each item retains its supporting request span, preventing an uploaded file or a scenario label from becoming an invented obligation. We distinguish whole-deck counts from requirements about a particular section, and mandatory requirements from preferences. For supported whole-deck count constraints, the actual page count determines the verdict and overrides the model judgment. Other criteria use the appropriate textual or visual evidence.

The hard-constraint rate (HCR) is

$$\mathrm{HCR} = \frac{n_{\mathrm{pass}} + 0.5 n_{\mathrm{partial}}}{n_{\mathrm{pass}} + n_{\mathrm{partial}} + n_{\mathrm{fail}}}, \tag{1}$$

where counts include applicable hard constraints only. Preferences are summarized separately. A failed hard constraint produces an explicit instruction-failure record even when most other items pass.

4.3 CONTENT QUALITY

Content evaluation examines coverage, explanatory depth, and faithfulness. Coverage specifications draw on explicit requests and task-relevant source material. Source-derived points are evaluation criteria grounded in the materials, rather than a claim that the user explicitly listed every required fact. Items receive *covered*, *partial*, or *missing* verdicts. Missing a required item activates the content branch of the failure analysis; partial coverage receives partial credit. Content-branch blocking failures are restricted to items whose source quotation is drawn directly from the task request or from explicitly specified source material.

Depth distinguishes whether a content module is present (L0), provides concrete information (L1), and is appropriate for the task (L2). Faithfulness compares claims with the sources and assigns *supported*, *derived*, or *unsupported* labels. Thus, a deck may cover a topic while misstating a source fact, or remain faithful to what it includes while omitting important content. Tasks without source materials still receive coverage and depth assessment.

Let r_0, r_1, r_2 be the depth pass rates. The content measures are

$$C = \frac{n_{\text{covered}} + 0.5n_{\text{partial}}}{N_{\text{applicable}}}, \qquad D = .20r_0 + .35r_1 + .45r_2, \tag{2}$$

$$B = .70D + .30C, \qquad S_{\text{content}} = \begin{cases} .60B + .40F, & \text{with a source-faithfulness score,} \\ B, & \text{otherwise,} \end{cases} \tag{3}$$

where F is the faithfulness score. Appendix D details the component mapping. The evidence example in Appendix K shows how the same frozen items distinguish outputs that use different lesson materials.

4.4 VISUAL PRESENTATION AND FAILURE DIAGNOSIS

Visual assessment combines four binary defect checks, six design criteria, and three aesthetic criteria (Table 2). Let H be the mean defect-check pass rate, V the mean of six design ratings normalized by 2, and A the mean of three aesthetic ratings normalized by 3. The page score is

$$S_{\text{page}} = .35H + .40V + .25A. \tag{4}$$

The deck visual score averages its page scores. We also count affected pages and decks for each defect. This measures a different property from average quality: how often generated information becomes clipped, obscured, incomplete, or visually degraded.

For the main failure analysis, a deck is flagged when it has a hard instruction failure, a missing required-content item, or at least one page with cropping or occlusion. These events are the operational *blocking failures* introduced in Section 1. All four visual categories remain in the diagnostic output; cropping and occlusion define the visual branch of this analysis. The protocol reports the three dimensions and failure evidence without requiring a single cross-dimension score: each dimension receives a score, and blocking failures are reported alongside it as explicit flags with page locations.

5 EXPERIMENTS

We study three questions: which important failures persist in commercial generation systems; how well existing quality scores reflect them; and whether the protocol exposes the effects of workflow improvements. Validation then examines the judgments on which these findings depend.

5.1 FAILURE PATTERNS ACROSS SIX COMMERCIAL SYSTEMS

Kimi, Doubao, Skywork, MiniMax, Manus, and AnyGen are publicly available commercial presentation-generation services; access dates and configurations appear in Appendix A. Figure 2 reports quality scores alongside failure counts. SlideGate detects 124 decks with at least one blocking failure. Missing required content is the most frequent category (71 decks), followed by cropping or occlusion (60) and hard instruction failures (33). These categories capture substantially different outputs: 88 decks exhibit only one category, while 36 exhibit multiple categories (Figure 8). Joint diagnosis therefore adds coverage beyond any single failure type.

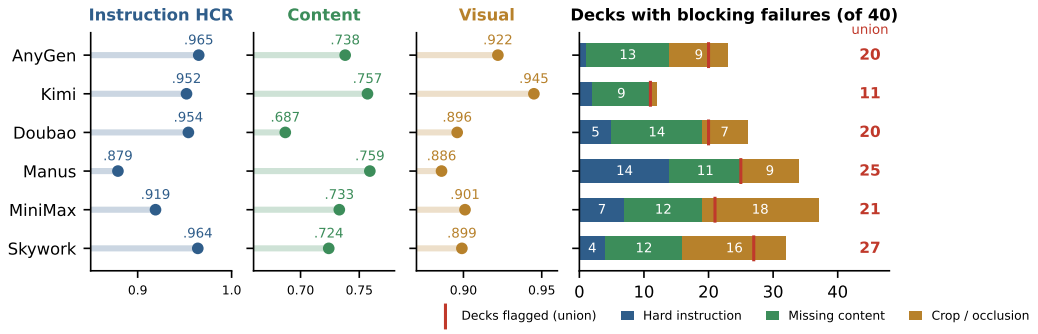


Figure 2: Six-system results. Left: instruction, content, and visual scores. Right: decks with a hard instruction failure, missing required content, or cropping/occlusion; the red number is the union, counting each deck once. Score gaps are small; the failure mix is not.

Different systems expose different capability gaps. Manus has the lowest HCR and the most instruction-failure decks, despite a content mean close to Kimi's (.759 versus .757). Conversely, AnyGen leads instruction following without leading content quality. MiniMax has the most decks with cropping or occlusion (18), whereas Skywork has the most decks with any blocking failure (27). These profiles identify distinct improvement targets: preserving explicit constraints, selecting the required content, and maintaining information readability after rendering.

Visual failures can extend beyond an isolated page. Of the 60 decks with cropping or occlusion, 19 have such defects on multiple pages, and one has affected pages throughout much of the deck (8 of 15 pages). Across all four visual defect categories, 326 pages are affected. The broader visual count includes content-integrity and image-quality issues as well as cropping and occlusion (Appendix G). The distinction matters: a system can achieve an attractive average appearance while leaving particular code snippets, chart labels, or teaching explanations unreadable. Section 6 examines these information-bearing failures directly.

5.2 Do Existing Quality Scores Reflect These Failures?

We score the same outputs using six schemes based on published evaluation criteria, applying each framework's released prompts or calculators where they accept generated decks. The comparison includes quality ratings, a defect-based aggregate, and computational aesthetics, as detailed in Appendix E. Most of these scores do not see the request or the source materials, so the analysis asks how far quality scores transfer to failure screening, not whether they serve their original purposes. We first examine score distributions by detected failure status, then test whether a threshold on a quality score can identify individual failure-containing decks.

Some scores separate the groups, but the gaps are small. The initial analysis compares decks with at least one detected blocking failure against those without a detected failure. SlideAudit, PPTBench, and AutoPresent assign significantly lower scores to the failure group, whereas PPTEval, SlidesGen, and LecEval do not (Figure 3). The differences are modest in absolute terms: for SlideAudit, the median score is .985 in the failure group and .990 otherwise. Existing scores thus carry some failure signal; the question is whether it suffices to locate individual failing decks. A non-significant overall score may also contain useful components; PPTEval's content subscore shows a difference in a separate content-only analysis (Appendix F).

Group separation does not ensure useful individual screening. For threshold-based screening, positives contain a detected failure, and negatives have evaluable judgments in all three dimensions with no detected failure. Each task and all its system outputs are held out together. On the remaining tasks, we select the strictest quality-score threshold attaining at least 90% failure recall, then apply it to the held-out task. AUROC intervals use task-cluster bootstrap. Appendix F specifies the statistical protocol and group definitions.

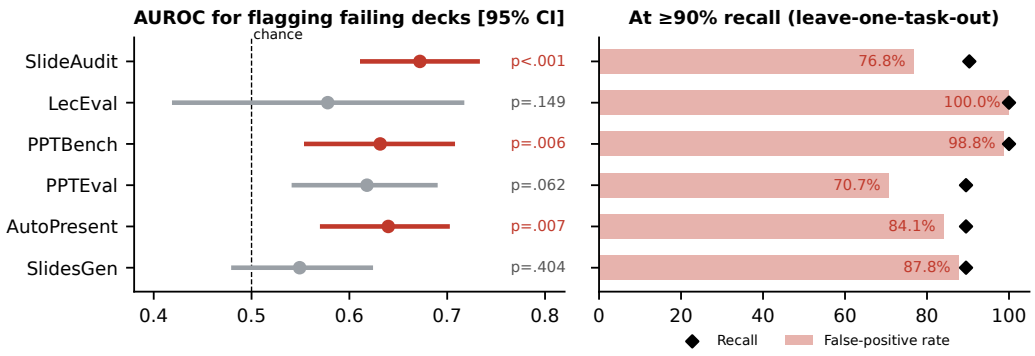


Figure 3: Quality scores as failure detectors. Left: AUROC with 95% task-bootstrap intervals; the dashed line is chance, and red marks schemes whose initial group test has $p < .05$. Right: false-positive rate when the threshold is set for at least 90% recall under leave-one-task-out; diamonds mark the achieved recall.

Table 3: Same-backbone workflow comparison. Direct and Multi-stage are unweighted means of task scores; Paired change is the mean of per-task differences over tasks scored under both workflows, so it can differ from the difference of the two means. Intervals are task-paired; p values use the Wilcoxon signed-rank test.

Measure	Direct	Multi-stage	Paired change	p	95% interval
Instruction HCR	.822	.973	+.149	.0007	[.075, .222]
Content	.677	.817	+.137	$< .001$	[.084, .191]
Visual page score	.905	.926	+.011	.59	[−.016, .038]

SlideAudit has the strongest ranking signal in this comparison, with AUROC .672. Yet detecting 112 of 124 failure-containing decks also flags 63 of 82 negatives: 90.3% recall with 76.8% FPR. PPTBench and AutoPresent reach FPRs of 98.8% and 84.1%. Thus, even the scores that separate the initial groups leave users inspecting most decks without detected failures. Across schemes, FPR ranges from 70.7% to 100.0% at the selected operating points. Quality ranking and failure diagnosis are related but distinct evaluation objectives; concrete item and page evidence explains what needs attention.

5.3 How Does a Generation Workflow Change the Failure Profile?

We compare direct generation and a multi-stage workflow using the same Claude Opus 4.5 backbone and the full 40-task suite. The multi-stage workflow converts the request and materials into constraints and usable assets, produces a deck plan assigning content to each slide, renders individual pages, and performs a post-generation check; a failing page may be revised at most once. This experiment asks whether the evaluation captures the capabilities affected by the overall workflow.

Instruction HCR and content improve substantially (Table 3), while instruction-failure decks fall from 15 to 1. Visual improvements are more selective. On matched tasks, cropped pages decrease from 10 to 9 and occluded pages from 7 to 4. The average within-deck fraction of pages with either defect falls from 4.16% to 2.86%, although the paired difference is not significant. The number of decks with either defect changes from 11 to 12, showing that fewer affected pages does not necessarily mean fewer affected decks. These results distinguish clear gains in constraint execution and content organization from remaining rendering weaknesses. Appendix I gives the full visual breakdown; the comparison measures the workflow jointly rather than attributing gains to an individual stage.

5.4 Validation of the Evaluation Dimensions

Instruction judgments against programmatic references. For explicit whole-deck page-count and aspect-ratio constraints, the protocol applies deterministic code rules that compute the verdict directly, ensuring such objective constraints are not left to model interpretation; Appendix C describes the count-verification and rule-augmentation procedure. Semantic requirements—for example, retaining a template or including a requested section—use the frozen criterion with its verbatim request quotation and the output evidence most relevant to that requirement. Every accepted instruction item is required to carry a supporting quotation extracted directly from the request text, preventing criteria not stated in the request from entering the specification.

Content judgments against independent annotations. Two annotators, blinded to system identity and evaluator output, label the same content-depth items. On binary L0–L2 judgments, evaluator agreement with the two annotators is 80.7% and 85.6%. Separate reliability analyses report 97.3% rerun agreement, Kendall correlation .867 between evaluator-induced system rankings, and 90% target-item detection under controlled content perturbations. Appendix H specifies the annotation protocol and label mapping.

Visual quality against designer preferences. Three professional designers provide anonymous pairwise preferences. On the six commercial systems, the designer-derived Bradley–Terry ranking correlates with mean visual scores at Spearman $\rho = .829$ ($p = .058$, exact two-sided permutation test). In a separate pairwise comparison collection limited to the six commercial systems, human and evaluator preference directions agree on approximately 70% of non-tied pairs. The appearance-focused preference protocol explains why content scores correlate weakly with these preferences ($\rho = .029$): content verification measures information that visual preference alone does not establish.

Specification validity. To test whether the auto-generated content items capture genuinely required information, two additional annotators judge each of the 270 frozen content items as *required* or *not required*, seeing only the task request and source attachments without any generated slides. Across the 33 tasks with content specifications, 90.0% of items are judged required by both annotators (inter-annotator $\kappa = .65$). Appendix H.4 reports the breakdown by source type.

Specification consistency. The three-candidate consolidation yields frozen specifications with span-level $F_1 = .974$ against the individual candidates; the frozen specification matches across systems on all 40 tasks, and repeated judgments agree on 95/96 items. These checks support consistent execution of the shared protocol. Appendix J reports score-weight and visual-penalty analyses.

6 Failure Mode Analysis

Rendered defects can remove the information a slide is meant to communicate. A MiniMax deck explaining Vue3 reactivity has overlapping code on page 7 and clipped dependency-structure content on page 12 (Appendix L). The defect is not merely decorative: the code and dependency relations carry the technical explanation. Figure 4 applies the same page-level checks to one page of each defect type and to a high and a low design rating.

A physics-teaching request asks Skywork to preserve an uploaded lesson while applying a supplied template. Cropping or occlusion is detected on eight of its 15 pages, including teaching content clipped at the bottom of page 3; other affected pages include derivations and formulas (Appendix L). Such pervasive defects are also penalized by the comparison scores, which rank this deck fifth or sixth within its task.

Instruction failures extend beyond slide count. For the SMART GOAL talk, Manus exceeds the requested 5–10 slides by producing 12. Its HCR is .50. In a board-level strategy task, the request enumerates eight required sections, from strategic background to decisions requested of the board; MiniMax omits required sections (HCR .72) while producing a complete-looking deck. Across the ledger, missing content requirements (40) and required sections (16) are more frequent than page-

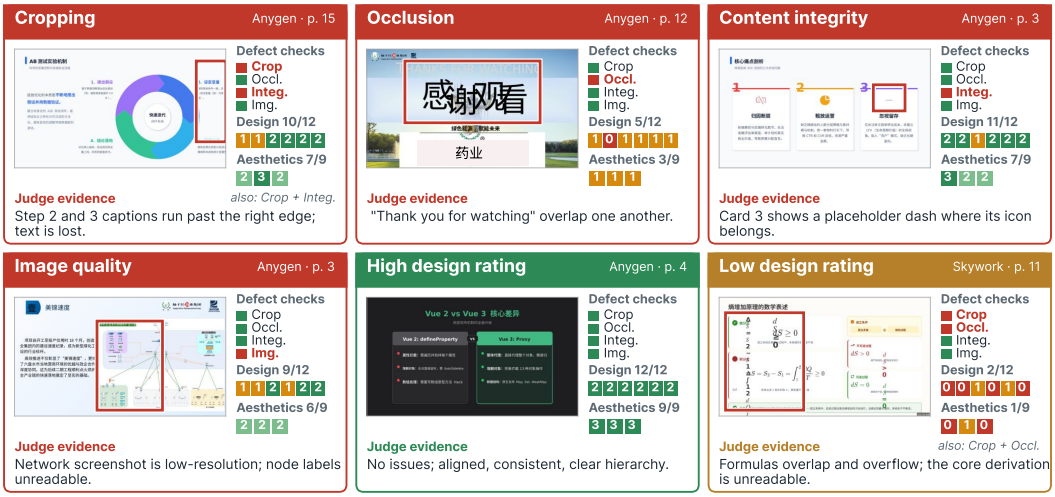


Figure 4: Visual checks on real output pages. Each card reports the four binary defect checks, six design ratings (0–2), and three aesthetic ratings (0–3). The first four pages each fail one defect check; the last two are a high and a low design rating. Where a page fails more than one check, the card says so. Pages are unmodified; boxes and labels are author annotations.

count violations (10). Verifying these requirements needs the frozen request spans rather than an impression of overall quality.

A complete-looking deck can use the wrong material. In a training task requiring lesson 2, Doubao instead uses lesson 1. Applying the same seven coverage items to two outputs gives Manus six covered and one partial verdict (coverage .929), but Doubao one partial and six missing verdicts (.071). Appendix K locates the source and output pages.

7 Discussion and Limitations

The benchmark exposes three practical improvement targets: preserve explicit constraints, retain and substantiate task-relevant information, and keep that information readable after rendering. The cases illustrate why these targets matter to use: finding a lost derivation, an unreadable code segment, or a wrong-source lesson requires more than comparing average design scores. They motivate evaluation that connects quality changes to concrete failure patterns; this study does not measure repair time.

Several limits guide interpretation. Criterion generation and evaluation use model-based processing verified through rule checks, rerun consistency, and human annotation; Appendix H reports the corresponding reliability measures and their scope. Visual preference validation addresses system-level quality ranking. We will release the tasks, attachments, frozen specifications, outputs, and evaluation code to support broader evaluation and extensions.

8 Conclusion

SlideGate evaluates presentation generation through task-specific verification and rendered-page diagnosis. Across six commercial systems, it detects 124 decks with hard instruction failures, missing required content, or cropping and occlusion. These failures affect different outputs and persist in some highly ranked decks. Several comparison scores separate failing and non-failing decks on average, but the gaps are small, and even the separating schemes incur substantial false positives in high-recall screening. The same-backbone experiment shows clear instruction and content gains but selective visual improvements. Together, these findings make constraint execution, source-aware content, and information-preserving rendering concrete targets for progress in presentation generation.

AI USE STATEMENT

Generative AI was used within the evaluation protocol to construct task specifications, generate experimental presentations, and judge outputs. It was also used to polish the manuscript. The authors checked the analyses, citations, and final text.

ETHICS STATEMENT

Human annotation and designer evaluation were conducted by company employees as compensated work under their regular salaries. Participants knew that their judgments would be used for research and agreed to that use. The authors confirmed the institutional ethics-review requirements for this annotation activity. We will release the task data, attachments, generated outputs, and evaluation artifacts, with identifying participant information excluded. The benchmark is intended to reveal shortcomings in generated presentations and support system improvement. Reported model judgments describe the specified tasks and evaluation criteria; they should not be used to infer the quality of an individual's work or to certify suitability for unrelated applications. Commercial-system comparisons are tied to the evaluated outputs, since services can change over time.

REPRODUCIBILITY STATEMENT

Section 4 defines the evaluation dimensions and score computation. Appendices A–C provide the instance schema, criteria, specification procedure, and rule checks; Appendix E specifies the comparison components. Appendix F gives the group definitions, task-cluster bootstrap, and held-out threshold selection. Appendices H–J document validation, workflow comparisons, and aggregation analyses. The accompanying artifacts include the original evaluation prompts, statistical figure data, and plotting code. We will release the tasks, attachments, frozen specifications, system outputs, and evaluation scripts to support rerunning the analyses on fixed artifacts and evaluating additional systems. An anonymized supplementary repository containing the frozen tasks, specifications, per-item and per-page judgments, and evaluation code is provided for review at https://anonymous.4open.science/r/slidegate.

REFERENCES

Xin-Sheng Chen, Jiayu Zhu, Pei-lin Li, Hanzheng Wang, Shuojin Yang, and Meng-Hao Guo. PresentBench: A Fine-Grained Rubric-Based Benchmark for Slide Generation. *arXiv preprint arXiv:2603.07244*, 2026. URL https://arxiv.org/abs/2603.07244.

Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yale Song. DOC2PPT: Automatic Presentation Slides Generation from Scientific Documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. URL https://arxiv.org/abs/2101.11796.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and Trevor Darrell. AutoPresent: Designing Structured Visuals from Scratch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2501.00912.

Zheng Huang, Xukai Liu, Tianyu Hu, Kai Zhang, and Ye Liu. PPTBench: Towards Holistic Evaluation of Large Language Models for PowerPoint Layout and Design Understanding. *arXiv preprint arXiv:2512.02624*, 2025. URL https://arxiv.org/abs/2512.02624.

Daesik Jang, Morgan Lindsay Heisler, Linzi Xing, Yifei Li, Edward Wang, Ying Xiong, Yong Zhang, and Zhenan Fan. DECKBench: Benchmarking Multi-Agent Frameworks for Academic Slide Generation and Editing. *arXiv preprint arXiv:2602.13318*, 2026. URL https://arxiv.org/abs/2602.13318.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. FollowBench: A multi-level fine-grained constraints following benchmark for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pp. 4667–4688, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.257. URL https://aclanthology.org/2024.acl-long.257/.

Yukyung Lee, JoongHoon Kim, Jaehee Kim, Hyowon Cho, Jaewook Kang, Pilsung Kang, and Najoung Kim. CheckEval: A reliable LLM-as-a-judge framework for evaluating text generation using checklists. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 15771–15798, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.796. URL https://aclanthology.org/2025.emnlp-main.796/.

Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. WildBench: Benchmarking LLMs with Challenging Tasks from Real Users in the Wild. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2406.04770.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://arxiv.org/abs/2303.16634.

Michael Ofengenden, Yunze Man, Ziqi Pang, Liang-Yan Gui, and Yu-Xiong Wang. PPTArena: A Benchmark for PowerPoint Editing. *arXiv preprint arXiv:2512.03042*, 2025. URL https://arxiv.org/abs/2512.03042.

Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy X. R. Wang. D2S: Document-to-Slide Generation Via Query-Based Text Summarization. *arXiv preprint arXiv:2105.03664*, 2021. URL https://arxiv.org/abs/2105.03664.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. Benchmarking complex instruction-following with multiple constraints composition. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2407.03978.

Yunqiao Yang, Wenbo Li, Houxing Ren, Zimu Lu, Ke Wang, Zhiyuan Huang, Zhuofan Zong, Mingjie Zhan, and Hongsheng Li. SlidesGen-Bench: Evaluating Slides Generation via Computational and Quantitative Metrics. *arXiv preprint arXiv:2601.09487*, 2026. URL https://arxiv.org/abs/2601.09487.

Joy Lim Jia Yin, Daniel Zhang-Li, Jifan Yu, Haoxuan Li, Shangqing Tu, Yuanchun Wang, Zhiyuan Liu, Huiqin Liu, Lei Hou, Juanzi Li, and Bin Xu. LecEval: An Automated Metric for Multimodal Knowledge Acquisition in Multimedia Learning. *arXiv preprint arXiv:2505.02078*, 2025. URL https://arxiv.org/abs/2505.02078.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.07641.

Zhuohao Jerry Zhang, Ruiqi Chen, Mingyuan Zhong, and Jacob O. Wobbrock. SlideAudit: A Dataset and Taxonomy for Automated Evaluation of Presentation Slides. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2025. URL https://arxiv.org/abs/2508.03630.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. PPTAgent: Generating and evaluating presentations beyond text-to-slides. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14402–14418, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.728. URL https://aclanthology.org/2025.emnlp-main.728/.

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. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.05685.

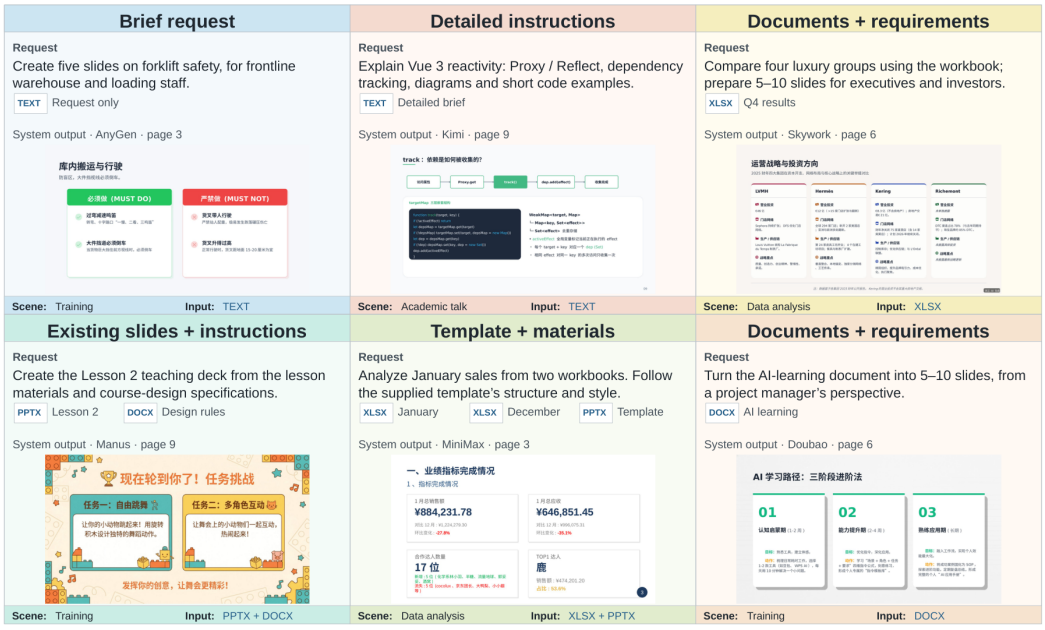
Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

A TASK INSTANCES AND REUSABLE ARTIFACTS

An instance records the request delivered to the generation system, its attachments, scenario and input-form metadata, frozen instruction and coverage specifications, and links between output pages and evaluation records. The formal benchmark request is the input used for evaluation; an earlier short user message is not substituted for it. Specifications are constructed once per task and reused across system outputs.

Table 4: Instance and evaluation schema. Identifiers connect task-level standards to output-level evidence.

Object	Required information
Task	Task ID; formal request; scenario; input form; attachment references
Source	Document ID; extracted text or rendered pages; source locations
Instruction item	Item ID; constraint; type; required/preference; supporting request quotation; judgment mode
Coverage item	Item ID; content point; request/source provenance; source quotation and location; relevance and importance
System output	Task ID; system ID; deck artifact; ordered rendered pages; page count
Item judgment	Item ID; verdict; evidence pages; explanation; verdict source
Visual judgment	Page index; defect checks; design and aesthetic ratings; localized evidence
Specification version	Stable item ordering and content hash



Representative task–output pairs. Requests are abridged and translated; file formats and rendered pages are retained.

Figure 5: Representative task–output pairs. Requests are abridged and translated; source slide images retain their original language.

Interfaces refer to the entry point used during the collection window; the paper does not evaluate versions released after this window. Frozen tasks, specifications, and rendered outputs are archived so that later interface changes do not affect the reported analyses.

Table 5: Evaluated commercial systems and access information. Access period corresponds to the collection window used for the outputs analyzed in the paper. Because these services evolve, capabilities are those available during the access window; later changes are not reflected in the reported outputs.

System	Interface used for evaluation	Access period	Outputs
AnyGen	Web slides workflow at anygen.io/product/slides (PPTX import and template supported)	2026-04 to 2026-05	40
Kimi	Web slides mode at kimi.com/slides (document upload and template supported)	2026-04 to 2026-05	40
Doubao	Web/PC client doubao.com/chat with the in-chat slide-generation entry (mobile app entry was being reduced during this period; not used)	2026-04 to 2026-05	40
Skywork	Web agent at skywork.ai/agent/zh-cn/slides (Skywork Super Agents slide expert)	2026-04 to 2026-05	40
MiniMax	MiniMax Agent PPT tools at agent.minimax.io (distinct from the Hailuo video product)	2026-04 to 2026-05	40
Manus	Web slides tool at manus.im/docs/features/slides in the web-slides-with-PPTX-export configuration (the native PowerPoint mode released 2026-07 was not used)	2026-04 to 2026-05	40
Total			240

B Criterion Definitions and Rating Anchors

B.1 Instruction Criteria

Count and format cover explicitly requested numerical specifications such as slide count and aspect ratio. Language covers requested output language. Layout and structure concern specified organization, such as a cover, outline, or conclusion. Mandatory content covers items explicitly named in the request. Template retention concerns requested preservation of visible template attributes. Additions and revisions concern requested changes to an existing deck. Requirements about the whole deck and requirements about a local section remain distinct.

A *pass* satisfies an applicable requirement, *partial* satisfies only part of it, and *fail* violates it. NA indicates that a judgment cannot be made from the available evidence; it contributes neither a pass nor a failure. Preferences remain visible in the report but are not included in HCR. Where no hard constraint is applicable, HCR does not provide a basis for comparing that task's instruction performance.

B.2 Content Criteria

Coverage items specify information to look for in the output. *Covered* means that the required information is present; *partial* means that only part is present; *missing* means that it is absent. A topic mention is not sufficient when the item requires concrete steps, facts, or activities. Source-derived coverage points retain the source passage and its task relevance. This separates the decision about what to evaluate from the subsequent judgment of whether it appears.

Depth has three levels: L0 asks whether the information module exists, L1 asks whether it contains specific information rather than generic statements, and L2 asks whether the treatment is appropriate for the request and setting. The levels assess adequacy, not an unconditional preference for more detail. A clear short explanation may satisfy L2 when the task calls for one.

Faithfulness uses three claim labels. *Supported* denotes a directly grounded or equivalent source statement. *Derived* denotes a valid calculation, summary, or inference traceable to the source. *Unsupported* denotes a claim that is neither supported nor a justified derivation. Derived claims are not counted as errors merely because their wording does not appear verbatim in the source.

Table 6: Visual criteria used by the page evaluator. Intentional layering is distinguished from overlap that impairs recognition.

Criterion	Definition
Boundary cropping	Content extends beyond a page or container and is clipped in a way that impairs recognition.
Occlusion/overlap	Text, diagrams, charts, or images block one another and impair reading; readable intentional layering is not a defect.
Content integrity	Incomplete content, empty frames, unreplaced placeholders, or missing chart components that hinder understanding.
Image quality	Visibly blurred, distorted, stretched, or otherwise degraded raster content. Absence of an image is not itself a defect.
Alignment	Consistency of margins, grids, and element alignment.
Spacing	Balanced gaps and density without crowding or unintended empty areas.
Component consistency	Consistent fonts, sizes, shapes, and emphasis for comparable components.
Legibility	Readable text and sufficient foreground–background contrast.
Type hierarchy	Clear, consistent differences between title, subtitle, and body levels.
Text flow	Appropriate leading, wrapping, and line breaks.
Hierarchy/reading path	Stable composition, clear emphasis, and a coherent reading order.
Visual language	Coherent typography, colors, and assets.
Overall impression	Apparent finish and visual completeness of the page.
Cross-page consistency	Coherence of the visual system across the deck.

B.3 Visual Criteria

The four defect checks are binary. The six design criteria use 0/1/2, progressing from a clear problem through partial adequacy to a satisfactory criterion. The three aesthetic criteria use 0–3. Exact prompts in the accompanying protocol files specify the judgment and output format. Defect-page counts are retained in addition to the mean page score.

C Rule-Augmented Specification and Judgment

C.1 Construction Procedure

The construction process separates proposing a criterion, checking its grounding, and judging an output:

1. Generate three candidate instruction or coverage specifications independently from the task inputs, without inspecting competing system outputs.
2. Consolidate candidates against the request and sources, merging equivalent requirements and preserving independently judgeable obligations.
3. Ask a critic to identify omissions, unsupported additions, duplicate weighting, incorrect requirement types, and ambiguous evidence.
4. Adjudicate the critique against the original inputs rather than resolving disagreement by majority vote alone.
5. Apply deterministic schema and provenance checks, order accepted items, assign stable identifiers, and compute a specification hash.
6. Evaluate every system output against that task's frozen items; validate returned identifiers and verdicts, apply supported rules, and aggregate.

C.2 Deterministic Count Rules

An exact count n is checked for equality; an explicit range $[a, b]$ includes both endpoints; a minimum or maximum imposes the corresponding one-sided bound. For an approximate count, the implementation uses $\max(1, \lceil .15n \rceil)$ as the tolerance. These checks apply to whole-deck counts.

Table 7: Implemented rule checks and their role in the protocol.

Stage	Rule	Purpose
Instruction grounding	Recover each supporting quotation from the request; reject empty or unsupported quotations	Prevent criteria inferred solely from common presentation conventions
Attachment interpretation	A file tag alone does not create an instruction to use or preserve its contents	Distinguish uploaded material from explicit user commands
Requirement typing	Validate criterion type; preserve mandatory/preference status and optionality	Avoid converting style preferences into mandatory constraints
Granularity	Remove duplicates; merge repeated deck-count ranges; distinguish total and local counts	Prevent duplicate weighting and incorrect count checks
Source grounding	Validate document identity, source quotation, and source location	Connect source-derived content points to inspectable material
Freezing	Sort items; assign IDs; hash the final content	Keep the standard fixed across systems
Judgment schema	Match verdict IDs to the frozen list; reject unknown or duplicate IDs and invalid labels	Prevent the judge from changing the scoring specification
Count verification	Compare the actual count to the explicit count or range	Resolve a supported numerical requirement deterministically
Evidence consistency	Apply evidence-availability and supported contradiction checks	Avoid treating unobservable properties as observed failures
Aggregation	Compute rates from applicable items using fixed mappings	Separate judgment generation from numerical scoring

A requirement such as one slide for each department is a local structural requirement and is not reduced to a whole-deck count check. The resulting count verdict takes precedence over the model verdict, with the actual and required values retained in the explanation.

Aspect ratio provides a separate deterministic validation reference when specified, and OCR-based language checks can provide supplementary evidence. The reported exact-reference validation combines count and aspect-ratio cases; the production override described above concerns page count. This distinction keeps the validation reference separate from the set of automatically overridden judgments.

C.3 Prompt Contracts

The following English summaries describe the instructions implemented in the accompanying original prompts.

Instruction extraction. Extract independently judgeable requirements from the request only. Preserve a contiguous supporting quotation for every item. Do not infer extra requirements from role, scenario, file type, or common practice. Do not force a fixed number of items.

Critique and adjudication. Check omitted explicit requirements, unsupported additions, duplicated obligations, requirement types, and the difference between total and local counts. Accept or reject critique by returning to the original evidence, not by blindly following the critic.

Coverage construction. Identify task-relevant information from the request and materials. Retain source passages, document identifiers, locations, and relevance. Distinguish core points used for coverage from contextual information.

Frozen-item evaluation. Return verdicts for the supplied IDs, with evidence pages and explanations. Do not introduce new criteria while evaluating an output.

Visual evaluation. Inspect the rendered page using the defined defect categories. Identify the affected location and the information whose recognition is impaired. Apply design and aesthetic criteria using their rating scales.

D Score Computation and Evaluation Configuration

The instruction aggregation uses Eq. 1. A hard failure is an applicable hard item labeled fail; partial satisfaction is not a full failure. A task specification is shared, while each output has its own applicability and verdicts.

Depth is $D = .20r_0 + .35r_1 + .45r_2$. Coverage is the applicable-item mean with covered/partial/missing mapped to 1/.5/0. Source faithfulness is

$$F = \frac{n_{\text{supported}} + n_{\text{derived}}}{n_{\text{claims}}}. \tag{5}$$

The composite follows Eq. 3. The within-content weights emphasize explanatory depth over mere mention, and the item-level verdicts remain available for alternative weightings; an unavailable source-faithfulness component does not imply that the claims have been verified. Source-faithfulness labels, coverage verdicts, and depth judgments remain separately inspectable.

The visual page score follows Eq. 4. Its hard component is the mean of four pass indicators, its design component the mean of the six ratings divided by 2, and its aesthetics component the mean of the three ratings divided by 3. The deck score averages page scores; defect-deck counts instead record whether any page has the specified defect. A page with both cropping and overlap is counted once in the cropping/occlusion union.

Table 8: Evaluation configuration. Each experiment uses its stated score components; workflow comparisons are paired within that experiment.

Component	Configuration
Instruction, content, and visual judgments	Claude Sonnet 4.6; frozen task specifications, item-level verdicts, and per-page scores; evidence provenance retained per record
Commercial content composite	Shared coverage combined with saved depth and source-faithfulness assessments from the same judge; component results retained
Rendered-page visual evaluation	Claude Sonnet 4.6; four defect checks, six design criteria, and three aesthetic criteria, aggregated as in Eq. 4
Six comparison scoring schemes	Quality scores from six representative frameworks on the same output collection; model scoring uses GPT-5.6-sol; computational aesthetics uses saved formula-based results
Same-backbone workflow comparison	Same Claude backbone and task suite; shared evaluation setup within the comparison; differences paired by task
Human validation	Dedicated annotation and preference collections matched to the judgment being validated

SlideGate's instruction, content, and page-level visual judgments are produced by Claude Sonnet 4.6. Comparison scores that call a model judge use GPT-5.6-sol, as specified in Appendix E. The commercial content components and workflow content scores are drawn from their respective evaluation runs, rather than used to rank the workflows against the commercial systems. The visual preference study likewise uses its paired scoring and annotation records. Preserving these experiment-specific inputs makes the reported comparisons reproducible without combining different scoring runs into a new experiment.

E Compared Scoring Schemes

PPTEval's design description considers overlap and contrast, AutoPresent's layout criterion explicitly considers overlap and boundaries, and SlideAudit includes corresponding defect categories. The comparison therefore examines the empirical behavior of these scores, rather than assuming that prior criteria omit all visual defects.

Table 9: Components used in the six-score comparison. Scores are oriented so that higher means better quality; normalization does not make them calibrated acceptance probabilities.

Scheme	Input and score	Evaluated criteria
SlideAudit	Rendered slides; one minus the mean number of flagged categories per page divided by 19	19-category design-defect taxonomy
LecEval	Rendered slides with the source text; mean of four released reward-model ratings, divided by 5; only the 151 decks with a content source	Teaching-quality criteria
PPTBench	Rendered slides; released generation-judge rating, divided by 100	Generation-quality scoring
PPTEval	Slide descriptions and deck text; content/design/coherence scores, divided by 5	Public multicomponent scoring criteria
AutoPresent	Rendered slides; text/image/layout/color ratings averaged and divided by 5	Reference-free evaluation
SlidesGen	Rendered slides; computational aesthetic features	Computational aesthetics

We apply the released scoring prompts, calculators, or weights to our rendered outputs: the PPTEval and PPTBench generation-judge prompts, AutoPresent's reference-free prompts, the SlidesGen-Bench aesthetics calculator, the SlideAudit taxonomy prompt from its paper appendix, and the released LecEval reward model. Prompt-based schemes share GPT-5.6-sol as a common judge. Components that require a framework's own inputs cannot be applied to a user request and its generated deck: SlidesGen-Bench's QuizBank requires its source documents and questions, its editability index requires native PPTX files, AutoPresent's reference-based metrics require its reference slides, and PPTBench's detection, understanding, and modification stages require its annotated files. Visual inputs are the same rendered pages for every scheme, with the text inputs listed in the table; pipeline-specific inputs such as element annotations or per-slide transcripts do not exist for commercial outputs; LecEval is scored on the 151 decks with a content source, and PPTEval scores come from an earlier run with the same judge. Cases are ranked within a scoring scheme and task. Ties occupy a rank interval, so a shared highest score is described as tied at the top.

F Statistical Protocol and Additional Score Analyses

Failure labels and groups. Instruction positives have a failed hard constraint; content positives have a missing required item; visual positives have cropping or occlusion on at least one page. Their union contains 124 decks. The initial group test asks whether scores differ between detected-failure and no-detected-failure groups. The latter describes detection status rather than confirmed absence of every possible failure. For screening, a negative additionally requires evaluable judgments in all three dimensions; the determinate-label analysis contains 124 positives and 82 negatives. These definitions explain the different purposes of the initial comparison and screening analysis.

Initial tests. We use two-sided Mann–Whitney tests for the six quality scores, reporting uncorrected p values as exploratory comparisons. Figure 6 displays each score on its own percentile scale to avoid equating numerical scales across methods. The group differences in the abstract refer to this initial grouping (SlideAudit $p = .0002$, PPTBench $p = .006$, AutoPresent $p = .007$, PPTEval $p = .062$, SlidesGen $p = .40$; LecEval $p = .15$ on its 151 scored decks). When positives must also have complete judgments in all three dimensions (104 positives, 82 negatives), SlideAudit, PPTBench, PPTEval, and AutoPresent show significant uncorrected group differences, whereas SlidesGen and LecEval do not. Nonsignificant comparisons are not tests of equivalence.

AUROC and threshold selection. Lower quality scores indicate higher failure risk. AUROC intervals use 3,000 bootstrap samples of tasks, retaining each sampled task's system outputs together. For leave-one-task-out screening, remove one task and all its outputs, choose the smallest quality cutoff that attains at least 90% recall on the remaining positive outputs, and flag held-out outputs

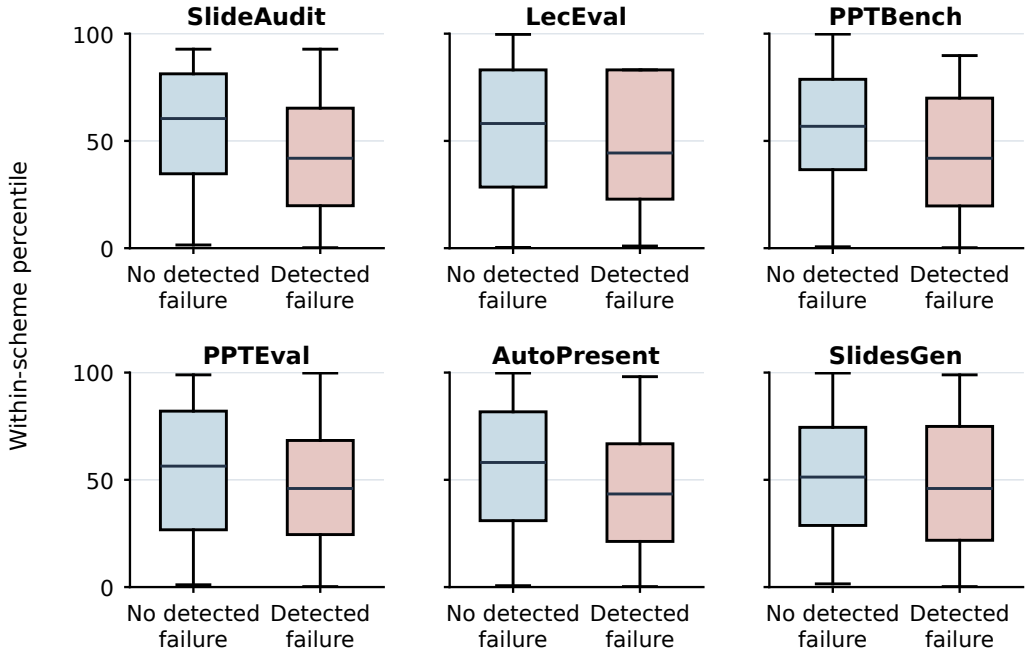


Figure 6: Initial score distributions in within-scheme percentile units. Boxes show medians and interquartile ranges, with standard 1.5-IQR whiskers; outlier markers are omitted. Tied scores share a midpoint percentile. A high percentile indicates a relatively favorable quality score, not an acceptance probability.

with scores at or below that cutoff. Tied scores can make training recall exceed the target. Held-out labels never select the cutoff. Aggregate predictions across folds to calculate recall $\mathrm{TP}/(\mathrm{TP}+\mathrm{FN})$ and FPR $\mathrm{FP}/(\mathrm{FP}+\mathrm{TN})$.

Table 10: Held-out confusion counts for the determinate-label screening analysis. LecEval covers only decks with a content source (81 positives, 52 negatives).

Scheme	TP	FN	FP	TN
SlideAudit	112	12	63	19
LecEval	81	0	52	0
PPTBench	124	0	81	1
PPTEval	111	13	58	24
AutoPresent	111	13	69	13
SlidesGen	111	13	72	10

Sensitivity and component scores. Requiring complete judgments in all three dimensions for positives as well yields the 104-positive, 82-negative subset reported above. PPTEval's content component also distinguishes a separate clean-versus-content-only comparison (.518 versus .465, $p = .041$), even though its overall score is nonsignificant in the initial union comparison. The different comparison groups preclude attributing that contrast solely to score aggregation. Together, the analyses support studying specific failure targets and score components rather than treating all evaluation outputs as interchangeable.

G Visual Failure Breakdown

Kimi's broader defect-page count is predominantly associated with content integrity, whereas MiniMax and Skywork have more cropping and occlusion. Accordingly, all-defect counts and the visual

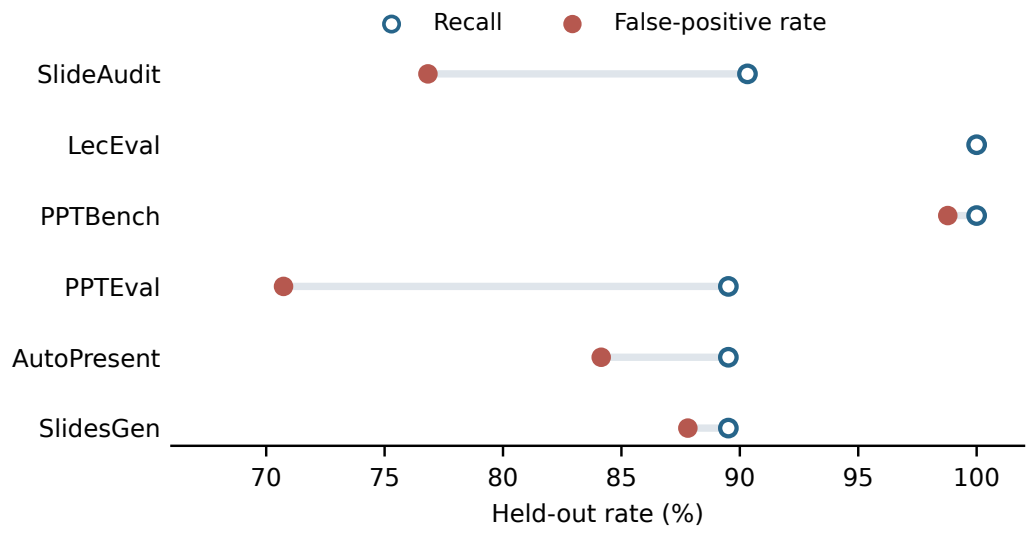


Figure 7: Held-out failure recall and false-positive rate at thresholds selected for at least 90% training recall. The high recall is accompanied by many false alarms.

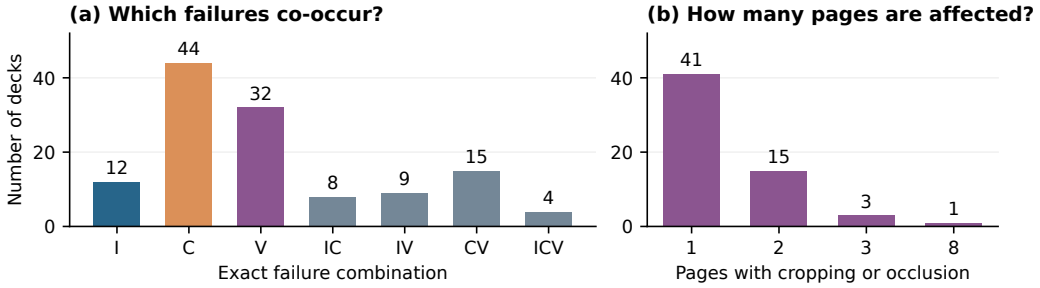


Figure 8: Failure structure. Left: mutually exclusive combinations of instruction (I), content (C), and cropping/occlusion (V) failures. Right: the number of affected pages in each of the 60 decks with cropping/occlusion. Counts are derived from saved item- and page-level judgments.

branch of the blocking-failure union answer different questions. Of the 60 H2 decks, 41 have one affected page, 15 have two, three have three, and one has eight. The count distribution quantifies concentration without assuming equal severity for all pages.

H Human Validation and Reliability

H.1 Programmatic Instruction References

The exact-reference validation compares model verdicts with directly computed page-count and aspect-ratio answers. The retained raw-judgment comparison has 46 items: 40 page-count and six aspect-ratio items. Overall agreement is 45/46 (97.8%); page-count failure recall is 5/6. Aspect-ratio items in this comparison contain no failures, so they do not establish failure recall for that type. The comparison uses the raw model verdict before any code override. The model evaluator has access to the actual page count, making this the most tractable subset of instruction items; the 5/6 failure recall confirms that deterministic override is warranted even for this favorable case, and code-derived verdicts take precedence in the production protocol.

Table 11: Visual defects by system. H4 includes all four binary visual categories; H2 includes cropping or occlusion. Per-category page counts can overlap.

System	H4 decks	H4 pages	H2 decks	Crop	Occlude	Integrity	Image
AnyGen	15	39	9	6	7	23	8
Kimi	11	46	1	1	0	45	1
Doubao	25	62	7	5	4	54	9
Manus	23	44	9	5	7	34	1
MiniMax	23	53	18	22	11	43	0
Skywork	28	82	16	17	17	64	1
Total	125	326	60	56	46	263	20

H.2 Content Annotation

Two annotators independently evaluate content items without seeing the system identity or model verdict. The annotation package contains 180 core L0–L2 items and 270 dynamic L0–L2 items. For binary L0–L2 agreement, human partial and fail labels are mapped to non-pass, and NA is excluded. The model agrees on 363/450 items with one annotator (80.7%) and 380/444 with the other (85.6%). These are item-judgment comparisons under supplied criteria, rather than a completeness test of the criteria themselves. Additional experiments assess content repeatability (97.3%), cross-evaluator ranking agreement (Kendall .867), and detection of targeted content perturbations (90%).

H.3 Designer Preferences

Three professional designers make anonymous pairwise judgments of visual appearance. A Bradley–Terry model converts these comparisons into a system ranking, which is compared with visual mean scores from the matching validation collection. On the six commercial systems, Spearman correlation is .829 ($p = .058$, exact two-sided permutation test). A separate blind comparison collection includes 126 scored, non-tied commercial-system pairs, with direction agreement on 89 (70.6%). The two analyses measure system ranking and pairwise preference, respectively.

H.4 Specification Validity

Two annotators who did not participate in the checklist construction independently label each of the 270 frozen content items as *required* or *not required*, seeing only the task request and source attachments. An item is *required* if the request cannot be considered fulfilled without it. Results are summarized in Table 12.

Table 12: Specification validity across 33 tasks and 270 content items.

	Count	%
Both required	243	90.0
Both not required	12	4.4
Disagreement	15	5.6
Inter-annotator κ	.65	

Of the 116 `query_requirement` items, 93.1% receive consensus-required labels. Of the 154 `source_fact` items, 87.7% receive consensus-required labels. The 12 consensus-not-required items are distributed across 9 tasks, with no single task contributing more than 3.

I Same-Backbone Workflow Analysis

The multi-stage workflow converts the request and materials into constraints and usable assets, plans slide roles and content, renders individual pages, and performs post-generation checks. A failing page can be revised once. Both workflows use the same Claude backbone and task suite. Paired score differences are computed on tasks with scores under both workflows. The comparison includes

all workflow changes jointly, including their computation, and does not isolate the contribution of one stage.

For visual defects, we match the saved outputs by task text, use valid page-level Boolean checks, and compute both counts and the within-deck fraction of affected pages. The latter is averaged over tasks, so long decks do not automatically receive greater weight. The two matched collections contain 429 and 433 evaluated pages, respectively.

Table 13: Visual-defect changes in matched workflow outputs. Values show direct → multi-stage. Task-average rates are means of within-deck defect-page fractions.

Defect	Affected pages	Affected decks	Task-average rate
Cropping	$10 \to 9$	$8 \to 8$	$2.79\% \to 1.79\%$
Occlusion	$7 \to 4$	$6 \to 4$	$1.61\% \to 1.07\%$
Cropping or occlusion	$16 \to 13$	$11 \to 12$	$4.16\% \to 2.86\%$
Any of four defects	$119 \to 127$	$29 \to 23$	$27.70\% \to 26.69\%$

Table 14: Paired changes in task-average defect rates, in percentage points (multi-stage minus direct). Intervals use 10,000 task bootstrap samples; two-sided Wilcoxon tests are exploratory and uncorrected.

Defect	Change	95% interval	p
Cropping	-1.01	$[-2.87, .67]$	.248
Occlusion	$-.54$	$[-2.49, 1.23]$	.574
Cropping or occlusion	-1.30	$[-4.08, .96]$	.623
Any of four defects	-1.02	$[-13.16, 11.12]$	.773

Some visual measures improve descriptively, but the rate intervals include zero and counts do not all move in the same direction. Reporting affected pages alongside affected decks distinguishes reduced defect concentration from elimination of failures across tasks.

J Aggregation and Ranking Analyses

Cross-dimension weights. A simplex grid with step .1 produces 66 weight settings and 22 distinct full rankings on the complete-score subset. Kimi ranks first under 40 settings, AnyGen under 21, and Manus under five. Minimum Kendall correlation with the equal-weight ranking is .067. This demonstrates sensitivity of these observed rankings to the aggregation policy; it does not establish that no application-specific weighting can be useful.

Ranking-direction conflicts. For two systems on the same task, a conflict occurs when one dimension orders them in the opposite direction from another. Observed conflict rates are 49.1% for vision/instruction, 40.2% for vision/content, and 41.3% for instruction/content. Task-based intervals are [35.8,63.2], [30.6,50.0], and [21.9,63.3]. Within-task permutation baselines are 49.9%, 50.0%, and 50.2%. The rates are descriptive and close to their baselines, so they are not evidence of a departure from independence.

Visual-penalty variant. A supplementary scoring variant maps four hard checks and six design ratings to ten normalized values z_j, and uses $\max(0, \sum_j z_j/10 - .8d/10)$, where d counts zero-valued criteria. This differs from the six-rating design component of Eq. 4. On the ablation's matched scoring collection, introducing the linear penalty expands the system-score range from .103 to .141 and its standard deviation from .052 to .071. Correlations with the two associated designer references are .952 and .943; multiplicative gating by .65 yields correlation .794. These results compare aggregation variants within that ablation and are not substituted for the main visual validation correlation.

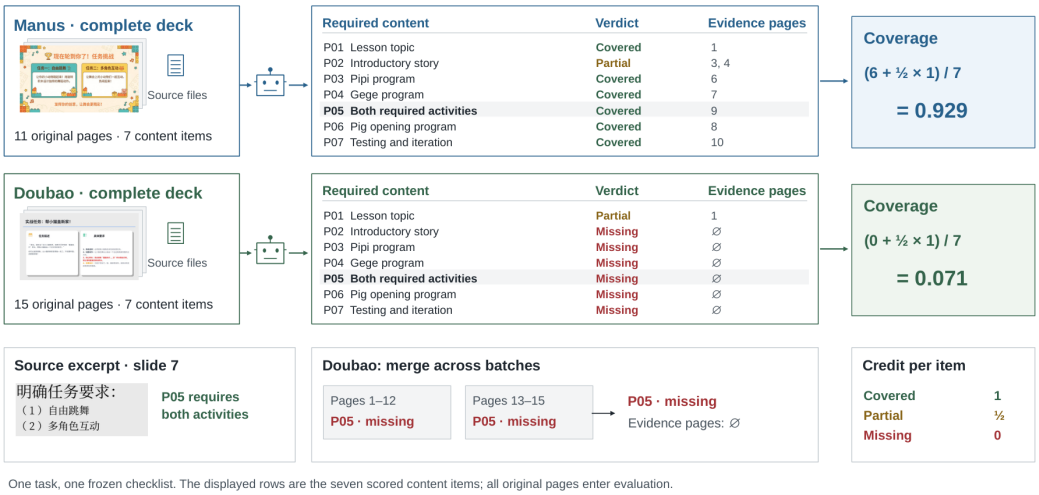


Figure 9: The same seven content items applied to two training decks. Manus receives six covered and one partial verdict, giving $(6 + .5)/7 = .929$ coverage. Doubao receives one partial and six missing verdicts, giving $.5/7 = .071$.



Figure 10: Source and output evidence for the requested lesson-2 activities. The relevant activities can be located in the grounded output, including Manus page 9, and are absent from the corresponding Doubao deck.

K SOURCE-GROUNDED CONTENT EXAMPLE

The example distinguishes an organized deck on a related lesson from coverage of the requested lesson. It also illustrates how a compound coverage criterion can be partially satisfied: a visible mention of one component does not establish that both requested activities are covered.

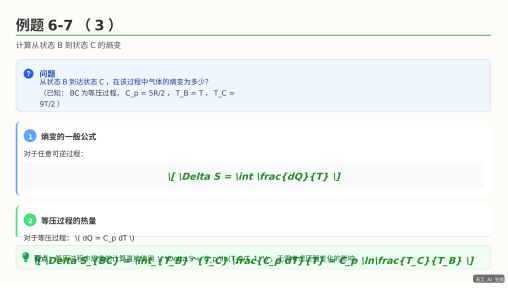
L Enlarged Visual Cases and Relative Score Positions

Table 15: Within-task rank of selected decks among six outputs, higher quality first. Intervals indicate ties; – marks decks without a LecEval score. Company intro fails a hard instruction; Payment survey misses required content. These are illustrative cases rather than an estimate of all methods' error rates.

Case	SlideAudit	LecEval	PPTBench	PPTEval	AutoPresent	SlidesGen
Company intro / Manus	1	6	1	1	1	3
Payment survey / AnyGen	1	1	1	3	1	2
SMART / Manus	5	6	2	2	3	4
Physics / Skywork	5	–	6	6	6	6
Vue3 / MiniMax	3–4	–	4	5	2	1



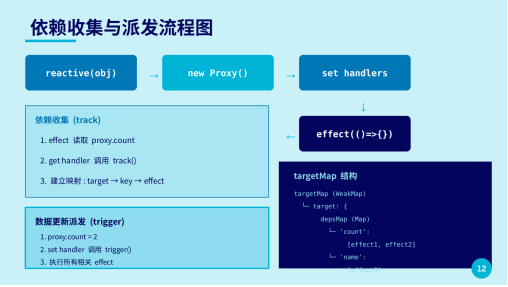
Physics, page 3: the teaching block and formula run off the bottom of the page.



Physics, page 9: the derivation and lower formula are clipped again.



Vue3, page 7: the final code lines overlap.



Vue3, page 12: the dependency diagram extends past its container and clips at the edge.

Figure 11: Enlarged pages for the physics and Vue3 decks in the rank table. The physics deck repeats the same clipping on two pages, so the defect is not an isolated boundary. The Vue3 deck shows overlap on the code example and clipping on the dependency diagram. The extensively clipped physics deck is ranked low by the comparison scores, whereas the Vue3 deck, with two affected pages, still ranks first under SlidesGen.