REDECK: STEP-LEVEL RENDER-GROUNDED REFINEMENT FOR DOCUMENT-TO-SLIDE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Document-to-slide generation is challenging because slides are dense editable artifacts that require both faithful content selection and precise spatial layout. Recent slide agents adopt iterative reflection, but typically follow a monolithic "one version, one feedback" loop: a slide or deck is rewritten, rendered afterward, and critiqued only at the turn boundary. This delayed feedback makes local failures such as overflow, overlap, clipping, and off-canvas placement difficult to attribute and repair. We propose **REDECK**, a step-level render-grounded refinement framework that decomposes slide revision into atomic edit actions and returns renderer-derived observations after each step, turning refinement into "one edit, one observation." To balance local repair with global quality, REDECK uses multi-granular feedback: step-level render feedback for spatial errors, a turn-level adaptive critic for semantic and design guidance, and a submission-level gate for hard layout validation. We further introduce DECKQUIZ, a benchmark that decouples content fidelity, spatial correctness, and design quality. Across GPT-5.4, Claude-4.6, and Gemini-3.1, REDECK obtains the highest mean scores among the evaluated slide-generation agents on all four DECKQUIZ modules, and ablations indicate that feedback timing and granularity are important for reliable slide refinement.

1 INTRODUCTION

Presentation slides are a central medium for communicating complex documents, including scientific papers, technical reports, and business analyses. Unlike plain text summaries, slides are dense visual artifacts: they must preserve source semantics, organize information into a coherent narrative, and arrange text, figures, charts, icons, and visual emphasis within a bounded two-dimensional canvas. Document-to-slide generation is therefore a coupled semantic-spatial problem, where content fidelity, layout validity, and visual readability must be optimized jointly.

Recent document-to-slide systems have largely followed a generate-then-render paradigm (Fu et al., 2022; Zheng et al., 2025; Ge et al., 2025; Liang et al., 2025; Zheng et al., 2026). Early methods attempt to produce a complete deck in one pass, for example by extracting an outline and rendering slides with predefined layouts (Liang et al., 2025). Since one-shot generation often fails to satisfy the many constraints of a polished deck, recent systems naturally introduce reflection: the agent renders a candidate deck, receives feedback from a critic, and revises the deck in the next turn, following the broader practice of LLM self-correction (Wei et al., 2022; Yao et al., 2023b; Madaan et al., 2023; Shinn et al., 2023; Zhou et al., 2024). However, this standard refinement loop is still *monolithic*: the agent commits a large slide- or deck-level rewrite, observes the rendered result only afterward, and then receives a holistic critique. In other words, existing agents mostly follow a "one version, one feedback" pattern.

This pattern is poorly matched to slide refinement. Unlike natural images, slides are editable artifacts with many small, information-bearing elements. A single flawed slide may contain high-level issues such as missing contributions, incorrect claims, weak narrative flow, or poor source grounding, while also containing low-level issues such as text overflow, element overlap, low contrast, unused space, clipping, and misalignment. Precisely describing all these issues in one global critique requires difficult alignment between visual regions, source content, and editable code. Even when the critique is correct, repairing many entangled issues at once is hard: the agent must decide which defect to fix first, infer which edit caused each failure, and avoid breaking already-correct regions. As illustrated

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.

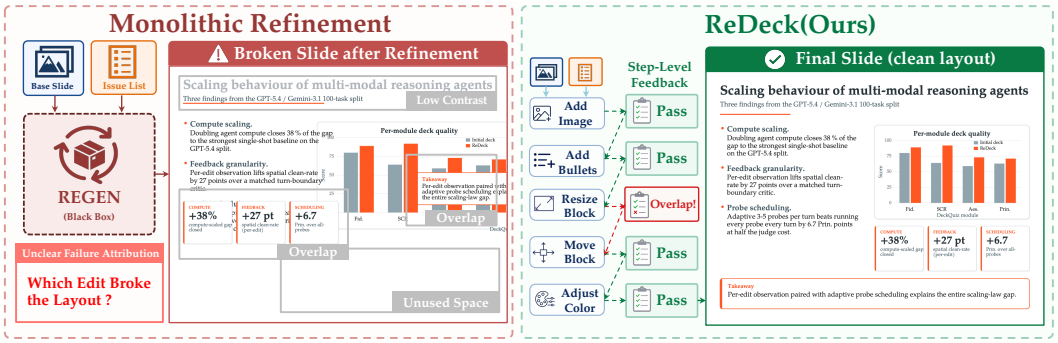


Figure 1: **Turn-level reflection vs. REDECK's step-level render-grounded refinement.** *Left*: conventional monolithic refinement commits large slide rewrites and receives feedback only after rendering the result, making overflow, overlap, low contrast, and off-canvas failures difficult to attribute to individual edits. *Right*: REDECK decomposes refinement into atomic edits and returns renderer-derived observations after each step, so the agent can fix local layout errors as soon as they appear while receiving global turn-level guidance.

in Fig. 1 left, after a monolithic rewrite, the deck may contain multiple spatial failures, but their causes are no longer attributable to individual edits.

This motivates a different refinement principle: feedback should be delivered at the granularity where the corresponding problem can be most reliably observed and fixed. For local layout errors, waiting until the end of a turn is unnecessarily delayed. If adding bullets immediately creates overflow, resizing a block immediately causes overlap, or moving an image immediately pushes content off the canvas, the agent should observe that rendered consequence before taking the next action. Such step-level feedback can reduce the critic's burden because each observation only needs to explain the consequence of the latest edit, and the editor's burden because the agent can fix one problem while the cause is still clear. At the same time, step-level layout feedback alone cannot judge whether the whole deck is faithful, complete, coherent, and well designed. Slide refinement therefore requires *multi-granular feedback*: fine-grained render-grounded observations for local spatial repair, and coarser turn-level reflection for global semantic and design direction.

We introduce REDECK, a *step-level render-grounded refinement* framework for document-to-slide generation. Instead of revising slides through monolithic rewrites, REDECK constrains the agent to a compositional action space of atomic edits, such as adding an image, inserting bullets, resizing a block, moving an element, or adjusting visual attributes. After each action, the rendering environment returns structured observations about the current visual state, including overflow, overlap, clipping, off-canvas placement, contrast issues, and key element geometry. This changes the refinement loop from "one version, one feedback" to "one edit, one observation", enabling the agent to ground its next edit in the actual rendered consequence of the previous one, as shown in Fig. 1 right.

To coordinate local repair with global presentation quality, REDECK implements multi-granular feedback with separated responsibilities. At the step level, deterministic render feedback reports immediate spatial facts but does not act as a global judge. At the turn level, an adaptive deck critic maintains a persistent issue list for higher-level concerns, including narrative flow, content completeness, factual correctness, source fidelity, and visual design. At the submission level, a validation gate ensures that the total hard-violation count does not increase relative to the session baseline before a turn is committed. This design lets each feedback source operate where it is most reliable: renderer observations handle local layout failures, the deck critic handles semantic and design direction, and the submission gate bounds the committed hard-violation count against the fixed session baseline.

To evaluate this framework, we introduce DECKQUIZ, a benchmark that separately measures content fidelity, spatial correctness, and design quality. This decomposition supports attribution of improvements to the corresponding refinement components rather than relying only on aggregated final-deck scores. Across GPT-5.4, Claude-4.6, and Gemini-3.1, REDECK achieves the highest mean scores among the evaluated slide-generation agents under a matched per-task LLM-call-count cap. Ablation

studies show that step-level render grounding substantially improves spatial correctness and stabilizes refinement, while the turn-level adaptive critic provides complementary gains in semantic and design quality. Shared-start controls, external-benchmark evaluation, a 45-task blinded human study, and PPTX transfer test the attribution, evaluator dependence, and format scope of the proposed refinement principle.

Contributions. 1) Step-level render-grounded refinement. We formulate slide refinement as a sequence of atomic edit actions, each followed by renderer-derived observations, enabling local layout failures to be detected and repaired at the moment they occur. **2) Multi-granular feedback for slide agents.** We combine step-level deterministic render feedback, turn-level adaptive reflection, and submission-level validation, assigning different feedback sources to the levels where they are most reliable and useful. **3) DECKQUIZ for component-level evaluation.** We introduce a benchmark that decouples content fidelity, spatial correctness, and design quality, enabling fine-grained attribution of improvements in document-to-slide generation.

2 RELATED WORK

Automated Slide Generation. Early document-to-slide work focused on content selection and layout from structured inputs (Fu et al., 2022; Sun et al., 2021; Bandyopadhyay et al., 2024). More recent systems frame slide creation as an agentic task: AUTOPRESENT (Ge et al., 2025), PPTAGENT (Zheng et al., 2025), SLIDEGEN (Liang et al., 2025), and AUTO-SLIDES (Yang et al., 2025) decompose the pipeline across multiple specialized agents, while SLIDETAILOR (Zeng et al., 2025) and ARCDECK (Ozden et al., 2026) further address personalization and narrative structure. Despite these advances, most systems evaluate and refine at a coarse turn-level granularity. DEEPPRESENTER (Zheng et al., 2026) is the most related, as it introduces rendered inspection within the refinement loop, but it does not validate layout integrity after each individual edit, allowing spatial errors to accumulate silently within a turn. Beyond slides, code-from-screenshot work in HTML/web settings (Si et al., 2025; Laurençon et al., 2024) addresses a closely related symbolic-to-rendered nonlinearity but operates on single-page artefacts without a multi-turn refinement contract, and the broader LLM-driven layout-generation literature (Gupta et al., 2021; Inoue et al., 2023; Feng et al., 2023) predicts element placements in one shot without a render-grounded repair loop.

Feedback and Self-Correction in LLM Agents. Iterative refinement through feedback is a central paradigm for improving model performance, building on the reasoning-and-acting trajectory established by chain-of-thought prompting and REACT (Wei et al., 2022; Yao et al., 2023b) and exemplified by frameworks such as *Self-Refine* (Madaan et al., 2023) and *Reflexion* (Shinn et al., 2023) which utilize verbal self-correction. Grounding feedback in external evidence has been further developed in works like CRITIC (Gou et al., 2024) and related self-correction studies (Huang et al., 2024b); execution-feedback variants ground the loop in compiler or test signals (Chen et al., 2024; Wang et al., 2024a; Huang et al., 2024a), while multi-agent SOP frameworks (Hong et al., 2024a; Wu et al., 2024) and software-engineering agent benchmarks (Jimenez et al., 2024; Wang et al., 2025) extend the same principle to richer environments. While environments like WEBARENA (Zhou et al., 2024), AGENTBENCH (Liu et al., 2024), and SWE-AGENT (Yang et al., 2024) demonstrate the necessity of real-time observations, and GUI / vision agents that ground actions in rendered screenshots (Hong et al., 2024b; Zheng et al., 2024) make the same point in the visual modality, slide agents often lack a specific rendered-state delta to capture immediate layout regressions. Existing systems predominantly rely on holistic or scalar reflection (Ge et al., 2025; Ozden et al., 2026; Zheng et al., 2026). In contrast, REDECK decouples feedback into step-level objective facts and turn-level semantic guidance to stabilize long-horizon refinement.

Benchmarks for Document Design and Layout. Accurate evaluation of generated presentations requires moving beyond generic text similarity (Guo et al., 2024; Ge et al., 2025; Zheng et al., 2025; Tanaka et al., 2023) toward metrics that preserve source fidelity and canvas integrity. Recent benchmarks such as PRESENTBENCH (Chen et al., 2026), SLIDESGEN-BENCH (Yang et al., 2026), ARCDECK (Ozden et al., 2026), AESLIDES (Pan et al., 2026), and DECKBENCH (Jang et al., 2026), alongside cross-format suites like BIZGENEVAL (Li et al., 2026), have introduced verifiable checks for layout dimensions, whitespace, visual imbalance, and reference-aligned content fidelity. These developments align with broader research on multi-constraint instruction following (Guo et al., 2025) and query-specific rubrics (Lv et al., 2026), emphasizing that evaluation signals should be

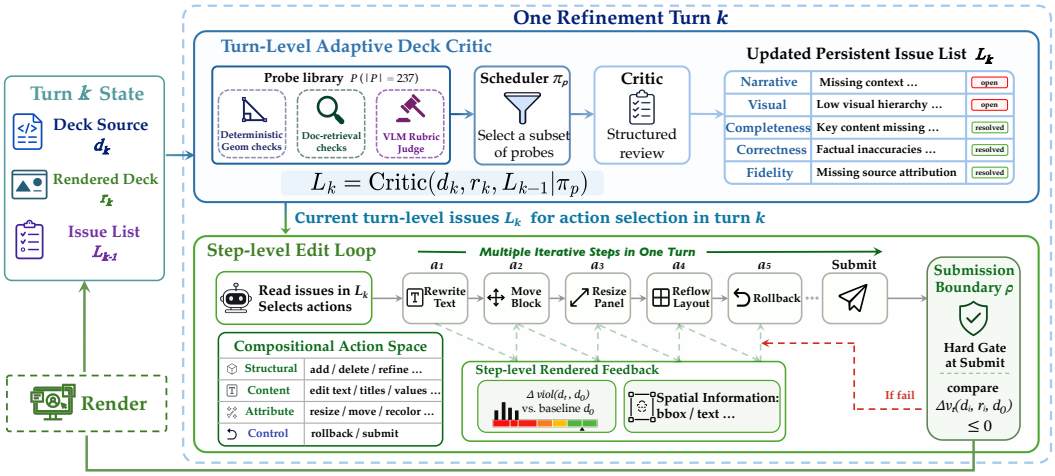


Figure 2: **REDECK pipeline.** Each refinement turn nests two feedback scales over a partially observable decision process (S, A, Ω). **Turn lane:** Adaptive Deck Critic uses a scheduler $\pi_{\mathcal{P}}$ to draw a probe subset from library $\mathcal{P}$ and updates the persistent issue list L_k. **Edit lane:** after every atomic action a_t from the compositional action space, the rendering environment returns render-state observation $o_t = (\Delta_{\text{viol}}(d_t, d_0), A_t)$ to the next reasoning step. Step-level render feedback is observation, not verdict; the only hard gate is the submission boundary ρ, comparing the rendered violations of the submitted deck against session-start baseline d_0.

instance-specific. Rooted in classical principles of perceptual organization and multimedia learning (Wertheimer, 1938; Mayer, 2009; Tufte, 1983; Lidwell et al., 2010), our proposed DECKQUIZ separates hard spatial-violation criteria from subjective design principles, while retaining the need to evaluate judge variance and bias (Zheng et al., 2023; Chiang et al., 2024; Kim et al., 2024; Qian et al., 2026; Wang et al., 2026).

3 METHOD

By splitting supervision into multi-granular feedback, consisting of an inner step-level render-feedback channel and an outer turn-level critic (Figure 2), REDECK reduces the feedback delay from an entire turn to a single edit.

3.1 PROBLEM MODELING AND COMPOSITIONAL ACTION DESIGN

The iterative generation of slide decks involves transitioning between a *semantic reasoning space* and a *rendered spatial space*. We model this process within a single turn as a sequence of discrete steps. Let t denote the inner step index. At step t, the dynamic state of the rendering environment is represented as the tuple: $s_t = (d_t, r_t, C_t)$, where d_t is the editable deck representation (e.g., HTML/CSS or structured presentation objects), r_t is the spatial geometry produced by the active rendering stack (e.g., browser DOM boxes for HTML or shape bounds for PPTX), and C_t contains compiler or export diagnostics (e.g., syntax errors or asset loading failures). Because the agent only edits d_t, it cannot directly perceive the physical layout r_t or diagnostics C_t without rendering. To avoid overwhelming the agent's context window with high-dimensional spatial coordinates and raw logs, the environment filters r_t and C_t to produce a compact, text-based observation $o_t \in \Omega$.

At the start of each refinement turn, the agent is also provided with a persistent issue list L. At each step t, the agent uses the active observation o_t alongside this static issue list L to select its next action $a_t \in A$.

To handle the highly nonlinear mapping from the source code d_t to the rendered geometry r_t (Si et al., 2025; Feng et al., 2023), REDECK constrains the agent to a compositional action space A of atomic edits. Rather than rewriting an entire slide in one step, the agent performs discrete modifications across three categories: *structural actions* for global element arrangements (e.g.,

adding or removing shapes, reflowing layout), *content actions* for updating semantic information (e.g., editing text, replacing figures), and *attribute actions* for precise geometric adjustments (e.g., resizing, repositioning).[1] This atomic design isolates the spatial consequences of each edit, helping the agent attribute newly introduced layout violations to a single, identifiable action. The action set is completed by `rollback`, which allows the agent to revert unsuccessful edits and use the `edit→observe→revert→re-attempt` loop as a deliberate local search reminiscent of tree-search reasoning (Yao et al., 2023a), and `submit`, which concludes the current turn subject to the validation gate.

3.2 Inner Loop: Step-Level Render Feedback

To help the agent see the direct results of its actions, ReDeck evaluates each atomic action a_t immediately, drawing the step-level supervision idea from process-reward modelling for reasoning chains (Lightman et al., 2024; Wang et al., 2024b) and porting it from *reasoning steps* to *rendering steps*. Instead of waiting until the end of a turn, the environment compiles the deck after every single edit and returns a step-level observation $o_t \in \Omega$ to the next step. This per-step loop creates a continuous cycle of editing, rendering, and observing, which helps detect layout errors before they accumulate. The step-level observation is defined as:

$$o_t = \big(\Delta_{\text{viol}}(d_t,\ d_0),\ A_t\big), \quad \Delta_{\text{viol}}(d_t,\ d_0) := |\mathcal{V}(d_t)| - |\mathcal{V}(d_0)|$$

where the violation delta Δ_{viol} measures the change in compiler failures and DOM-level errors (such as text overflow, overlap, and clipping) compared to the initial baseline d_0. To help the agent locate these errors, the environment provides this delta along with specific details, including the IDs and bounding boxes of the failing elements. The layout anchor A_t records the exact positions (x, y), dimensions $(w \times h)$, and text previews of the main elements, providing a clear numerical grid for planning while filtering out visual noise like page numbers. We explicitly exclude screenshots and subjective design scores at this step; step-level screenshots often cause the agent to hallucinate, while subjective scores are too inconsistent to guide individual edits.

During the editing process, this step-level feedback acts as a soft hint rather than a strict constraint. This soft policy is necessary because fixing a layout often requires going through temporary error states, such as a textbox temporarily overflowing before the agent can shrink its font. To bound the violation count at the end of the turn, ReDeck applies a hard check when the agent calls the `submit` action. This is controlled by a simple validation rule:

$$\rho(s_t, a_t, s_{t+1}) = \begin{cases} \mathbb{1}[\Delta_{\text{viol}}(d_{t+1},\ d_0) \le 0] & a_t = \texttt{submit}, \\ 1 & \text{otherwise.} \end{cases}$$

Since this rule uses the same fixed baseline d_0 as the step-level observations, the agent sees the submission criterion throughout editing. If $\rho = 0$, the submission is rejected, and control returns to the agent with the current observation for further repair while budget remains. Passing bounds the total violation count relative to d_0; it guarantees neither zero violations nor monotonic improvement from the preceding turn, and it does not certify semantic correctness.

3.3 Outer Loop: Turn-Level Adaptive Deck Critic

While step-level feedback focuses on individual edits, the agent still needs a broad, deck-wide view of which parts of the presentation are unsatisfactory. The Adaptive Deck Critic provides this view by evaluating both the slide source code d and its rendered images r at the start of each turn. This process updates a persistent issue list L defined as:

$$L \leftarrow \text{Critic}(d,\ r,\ L;\ \pi_{\mathcal{P}})$$

where $\pi_{\mathcal{P}}$ is the scheduling policy. Rather than giving free-form advice, the list L consists of structured entries. Each entry contains the issue category, the affected slide ID, a pointer to the evidence, and a status marker. The critic monitors five key categories: narrative flow, visual layout, completeness, correctness, and source fidelity. On each turn, the critic either adds new issues to L or

[1] At the tool level, these logical action types are implemented through a general-purpose `apply_edits` tool that performs targeted search-and-replace operations on the slide source code, along with `verify_layout`, `rollback`, and `submit` control tools. The agent is prompted to perform one logical edit per tool call; the three-way taxonomy describes the *intent* of each edit rather than distinct tool endpoints.

updates the status of existing ones to represent whether they are still present, resolved, or regressed. This structured approach ensures that resolving an issue requires a formal verdict from the critic rather than a simple self-report by the agent.

Probe library. To detect these diverse problems, the critic uses a library $\mathcal{P}$ of 237 single-purpose probes. These probes are based on published slide benchmarks, classic visual design principles, and multimedia learning guidelines. Each probe is implemented in one of three ways: geometric checks (such as finding text overlaps or font size errors), source-retrieval checks (using BM25 index lookups to verify factual claims), or rubric-based multimodal evaluations. This structure ensures that LLM critiques enter the loop as formal, typed entries rather than unstructured conversation. To maintain feedback stability, the scheduler $\pi_{\mathcal{P}}$ dynamically activates a small budget of three to five probes per turn. The scheduler ranks probes based on which slides were recently edited, previous regression history, and the need to recheck resolved issues. Running all 237 checks on every turn would create high judge variance, causing the critic to find inconsistent, minor issues that lead to conflicting edits and prevent the layout from converging.

3.4 The Full ReDeck Framework

At the initiation of each refinement turn k, the adaptive deck critic evaluates the current deck source $d_{k,0}$ and its rendered images to update the persistent issue list L_k, which remains static throughout the turn. The agent then executes an inner sequence of atomic edits in a thought–action–observation trajectory (Yao et al., 2023b), until it invokes the `submit` action to trigger the validation gate ρ. Upon a successful submission, the final compiled source d_{k,T_k} becomes the initial state for the next turn, setting $d_{k+1,0} = d_{k,T_k}$, while the gate reference d_0 remains the fixed session-start deck. A failed submission returns control to the agent for further corrective adjustments while budget remains.

To prevent optimization drift, the two scales operate under strictly decoupled write permissions: the turn-level critic holds exclusive authority to append new issues, bind localization evidence, or verify the resolution of existing entries in L_k, whereas the agent has read-only access to these fields and the high-frequency step-level channel serves strictly as an instantaneous spatial report without modifying L_k. This asymmetric information barrier prevents the agent from self-reporting success, keeping L_k an objective, environment-enforced target. The dual-scale synergy is essential for convergence: without the critic, the agent over-optimizes local layouts at the expense of deck-level coherence; without step-level spatial feedback, it edits the underlying source code blindly. We empirically evaluate this dual-scale topology in §4.

4 Experiments

4.1 Setup

Task and benchmarks. We target *scientific presentation generation* from full papers, the dominant evaluation setting in prior slide-agent work (Liang et al., 2025; Zeng et al., 2025; Zheng et al., 2026; 2025); external evaluation is in Sec. 4.5. DeckQuiz (ours) separates four evaluation modules: **ContentQuiz** (Fid., four QA types: contribution / method / experiment / limitation), **SpatialCheck** (SCR, VLM-judged clean-rate over four hard-violation families), **Aesthetics** (Aes., 0–5 ordinal over five design axes), and **DeckDesign** (Des., 0–3 Likert over five info-architecture principles in deck and slide layers; App. M); the four modules ask, respectively, *what* facts a deck conveys, *whether* they render cleanly, *how each slide looks*, and *how the deck organises information*. The in-domain split contains 100 papers stratified by discipline (CS/ML 25, Bio-Med 15, Physical-Sci 25, Econ/Finance 25, Soc-Sci/HCI 10) and complexity (30 short / 40 medium / 30 long); construction details, source freeze, and per-task call budget are in App. L. The data-overlap check against PresentBench (Chen et al., 2026) is described in App. G.

Models, comparisons, and protocol. We hold the agent prompt and tool budget fixed across GPT-5.4, Gemini-3.1, and Claude-4.6, and compare ReDeck against three published deck-level slide agents rerun end-to-end under matched per-task LLM-call-count cap: SlideGen (Liang et al., 2025), SlideTailor (Zeng et al., 2025) (code from `nusnlp/SlideTailor`), and DeepPresenter (Zheng et al., 2026) (code from `icip-cas/PPTAgent`; the most recent system, descended from PPTAgent (Zheng et al., 2025)). In-house refinement strategies (Single-Shot, Self-Refine (Madaan et al., 2023), Reflexion (Shinn et al., 2023), per-turn Screenshot) appear

Table 1: Performance comparison with published slide agents on DECKQUIZ (3 models × 100 tasks × 3 seeds) under matched per-task LLM-call-count caps. Best results per column are in **bold**.

Model	System	Fid. ↑	SCR ↑	Aes. ↑	Des. ↑
GPT-5.4	SlideGen	65.3	83.4	3.08	50.2
	SlideTailor	60.2	88.2	2.90	56.4
	DeepPresenter	76.8	66.8	3.47	55.1
	REDECK (T0, before repair)	80.4	64.1	2.95	62.8
	REDECK (ours)	**88.6**	**91.5**	**3.64**	**71.2**
Gemini-3.1	SlideGen	61.5	80.1	2.92	47.3
	SlideTailor	56.8	84.7	2.71	53.2
	DeepPresenter	72.4	63.5	3.31	52.0
	REDECK (T0, before repair)	76.8	61.2	2.81	59.4
	REDECK (ours)	**85.1**	**88.2**	**3.48**	**68.1**
Claude-4.6	SlideGen	63.2	81.6	3.01	48.6
	SlideTailor	58.4	86.2	2.82	54.7
	DeepPresenter	74.5	64.9	3.38	53.4
	REDECK (T0, before repair)	78.5	62.5	2.88	61.0
	REDECK (ours)	**86.8**	**89.6**	**3.55**	**69.5**

Table 2: Shared-start refinement on GPT-5.4 (100 tasks × 3 seeds). Both loops receive the same initial deck within each block.

Initial deck	Refinement	Fid. ↑	SCR ↑	Aes. ↑	Des. ↑
REDECK T_0	None	80.4	64.1	2.95	62.8
	DeepPresenter	78.0	72.0	3.10	63.0
	REDECK	88.6	91.5	3.64	71.2
DeepPresenter T_0	None	76.0	54.0	3.20	55.0
	DeepPresenter	76.8	66.8	3.47	55.1
	REDECK	84.0	87.0	3.50	65.0

in Sec. 4.3 as feedback-paradigm ablations sharing our parser, tool set, and prompt scaffolding so that the contrast isolates the feedback channel. Each (model, system, task) cell uses three seeds; main-table entries are means, with seed variability and task-paired inference reported in App. O. All systems run under a matched 150 LLM-call-count cap (App. N); token volume, cost, and latency are not matched and are reported separately in App. P. The main evaluation uses a fixed GPT-5.4 judge backbone for all systems (Zheng et al., 2023): ContentQuiz uses text-only input, while SCR, Aesthetics, and DeckDesign use rendered images (App. J). Internal DOM geometry supplies edit feedback and the submission gate; it does not produce the reported VLM-SCR scores.

4.2 MAIN RESULTS

Tab. 1 shows that REDECK achieves the highest mean score on all four DECKQUIZ modules for each agent backbone. GPT-5.4 task-paired inference supports all 12 comparisons after Holm correction, while the other two backbones serve as descriptive replications (App. O).

Because REDECK's initial draft is already strong on FID. and DES., we also evaluate both refinement procedures from the ReDeck and DeepPresenter initial decks (Tab. 2). The SCR advantage persists across both starts, with gains of 19.5 [16.3, 22.7] and 20.2 [17.1, 23.3] points over the DeepPresenter loop. The aesthetics difference from DeepPresenter T_0 is inconclusive (+0.03 [−0.10, 0.16]); full controls are in App. O. Component attribution is in Tab. 3, and a recorded repair trajectory is presented in App. C.

Table 3: Ablation on different feedback mechanisms. Each block isolates a single design choice; the shaded row is the REDECK configuration of Tab. 1. Per-turn convergence dynamics are in App. R, Fig. 7.

Variant	Fid. ↑	SCR ↑	Aes. ↑	Des. ↑
Feedback paradigm				
Initial generation (T0, no repair)	80.4	64.1	2.95	62.8
NL critique, per-turn (Self-Refine / Reflexion)	76.8	68.2	3.18	60.5
Screenshot critique, per-turn	78.2	78.4	3.41	65.1
REDECK*[default]*	88.6	91.5	3.64	71.2
Components				
Turn-level critic only	78.7	60.8	3.02	63.5
Action-level render feedback only	78.5	80.2	3.12	59.4
Per-step observation frequency — observation interval N between rendered observations				
N=4 (every 4 actions)	86.2	87.4	3.51	69.2
N=8 (every 8 actions)	83.5	82.1	3.38	67.5
Scheduling — subset selection over the 237-probe library $\mathcal{P}$				
All-probes-every-turn (no scheduling)	80.1	62.8	3.08	62.5

4.3 ABLATIONS

All ablations use GPT-5.4 with three seeds and shared initial decks. Tab. 3 compares turn-level natural-language and screenshot critique, isolates the two feedback components, varies the observation interval over $N \in \{1, 4, 8\}$, and contrasts adaptive scheduling with evaluation of all 237 probes at every turn.

The feedback-paradigm results show that both turn-level alternatives underperform REDECK, with the largest differences appearing in SCR and DES. The component ablations further show that the two feedback channels serve different roles: action-level render feedback substantially improves SCR but reduces DES. relative to T_0, whereas the turn-level critic provides a small design improvement without improving spatial correctness. Combining the two channels yields the best result on all four metrics, consistent with their complementary local and global functions.

Performance decreases consistently as the observation interval increases from N=1 to N=4 and N=8, indicating that feedback frequency affects both spatial correction and overall deck quality. Likewise, evaluating all probes at every turn performs close to T_0 and substantially below adaptive scheduling. Fig. 7 and App. R show that this setting raises additional visual findings whose subsequent repairs introduce geometric regressions. Together, these results indicate that frequent render observation and selective probe scheduling jointly improve refinement stability.

4.4 COST AND LATENCY

On the same 100-task GPT-5.4 split as Tab. 1, REDECK's end-to-end wall-clock is 3586 s (8.5× DEEPPRESENTER) and per-task USD cost is \$1.74 (14.5×). To test the effect of increasing the baseline's refinement budget, we increase DeepPresenter's post-T_0 call cap from 150 to 2175 on the shared REDECK initial deck. At this budget, DeepPresenter reaches 77.0/78.0/3.20/63.0 on FID./SCR/AES./DES., compared with REDECK's 88.6/91.5/3.64/71.2. A routed REDECK-Lite variant retains high-level planning and final submission on GPT-5.4 while sending recurrent execution and verification to a nano backbone; it costs \$0.28 per task and retains 89.8 SCR. App. P reports the cost, latency, and quality trade-offs, including the full call-budget sweep and its accounting boundaries.

4.5 ROBUSTNESS AND VALIDITY

External evaluation. To test whether the ranking depends on DECKQUIZ, we evaluate the same systems with DECKBENCH and SLIDESGEN-BENCH, which change the data and evaluator. Against DeepPresenter, REDECK improves DECKBENCH LayoutQ, DeckFaith, and DeckFid by 0.032, 0.023, and 0.022, respectively, and improves SLIDESGEN-BENCH QuizBank accuracy by 2.6 per-

centage points and computational aesthetics by 3.2 points. The protocols, common-set sizes, and system-level results are in App. H, including failures, AeSlides checks, and the DECKBench perplexities on which REDECK does not obtain the best result. These metrics remain related to slide quality, but are not derived from the REDECK probe taxonomy.

Transfer to PPTX. The refinement loop requires editable geometry and repeatable rendering rather than HTML APIs specifically. Replacing DOM queries with `python-pptx` shape geometry and browser rendering with LibreOffice improves a PPTX T_0 from $78.0/58.0/2.80/60.0$ to $83.0/82.0/3.30/66.0$ on the four DECKQUIZ modules (100 tasks $\times$ 3 seeds; App. I).

Human pairwise study. Because the probe library $\mathcal{P}$ and DECKQUIZ were built by the same authors, three independent annotators blindly compare REDECK with the strongest baseline, DeepPresenter, on 45 held-out GPT-5.4 tasks. Majority vote favors REDECK on overall presentation quality in $33/45$ cases (0.73, Wilson 95% CI $[0.59, 0.84]$) and on source faithfulness in $31/45$ cases (0.69, $[0.54, 0.80]$); the protocol and full counts are in App. E.

Internal-validity audits. Appendix audits examine pipeline artefacts through probe-taxonomy coverage (App. K), VLM-SCR calibration against non-author labels and a deterministic DOM cross-check (App. F), and per-principle-layer regression on DES. (App. Q). Subjective calibration and alternative judges are reported in Apps. D and J; Claude aesthetics versus DeepPresenter remains inconclusive. Title/DOI matching finds no paper overlap between DECKQUIZ and PRESENTBENCH.

5 CONCLUSION

We have presented REDECK, a step-level render-grounded refinement framework for document-to-slide generation. By decomposing slide revision into atomic edit actions and returning renderer-derived observations after each step, REDECK collapses the perception–action gap from an entire turn to a single edit, allowing the agent to detect and repair local layout failures at the moment they arise. The turn-level adaptive deck critic complements this with global design direction, while the submission gate bounds the hard-violation count relative to the session-start deck.

Experiments across three model backbones show that this dual-scale feedback achieves the highest mean scores among the evaluated slide agents on content fidelity, spatial correctness, aesthetics, and information architecture under the fixed judge. Ablations support complementary roles for step-level render feedback and the turn-level critic, while shared-start controls separate refinement performance from initial-deck quality.

More broadly, REDECK illustrates a general principle for iterative generation of persistent, inspectable artifacts: feedback should be delivered at the granularity where the corresponding problem can be most reliably observed and fixed. We believe this principle extends beyond slides to other domains where symbolic edits have nonlinear rendered consequences, including web pages, documents, diagrams, and interactive interfaces.

6 LIMITATIONS

REDECK requires editable geometry and repeatable rendering. We validate HTML/CSS and PPTX, but not PDF/OCR-only editing, Keynote, animations, masking, or font-fallback behavior. Native end-to-end comparisons also combine each system's initial generator with its refinement loop; the shared-start experiment isolates loop quality across two initial-deck sources but does not make the native generators identical.

The full system costs $14.5\times$ more and runs $8.5\times$ slower than default DeepPresenter. The call-budget sweep and REDECK-Lite clarify the effects of additional inference and model routing, but routing mitigates rather than removes the cost gap, and the sweep is not a dollar-matched comparison. The gate does not guarantee zero violations or monotonic improvement between turns. Finally, DECKQUIZ and the adaptive critic share related quality constructs. External benchmarks, non-author labels, a deterministic DOM cross-check, and blinded human preference address this concern without eliminating evaluator dependence. The 45-task human study and 50-deck subjective calibration remain moderate in size, and subjective visual harmony is still difficult to automate.

AI USE STATEMENT

Generative AI models are used in the slide-generation, refinement, and evaluation pipelines described in Sections 3–4 and Appendix J. An AI coding assistant was used to adapt the LaTeX source to the ICLR 2027 template, condense related work, repeated introductory material, and method descriptions, integrate author-supplied experimental summaries, audit claim consistency, and check compilation and layout. It also assisted in restoring the original manuscript's wording and explanatory structure while retaining corrections to claim scope.

ETHICS STATEMENT

Automated slide generation and refinement may introduce factual errors or misleading visual emphasis, even when layout quality improves. Generated slides should therefore be checked against source documents before use, especially in scientific or educational communication. Users should process only documents they are authorized to use, respect copyright, and avoid disclosing confidential or personal information to model providers. Human oversight remains necessary for responsible use of the generated content.

REPRODUCIBILITY STATEMENT

The refinement environment, feedback channels, and submission rule are described in Section 3; the initial-draft pipeline and rendering environment are documented in Appendix B. Appendices L, J, and N describe benchmark construction, judge settings, and call budgets. Task-paired inference and cost accounting are reported in Appendices O and P, and the human evaluation protocols are provided in Appendices D and E.

REFERENCES

Sambaran Bandyopadhyay, Himanshu Maheshwari, Anandhavelu Natarajan, and Apoorv Saxena. Enhancing presentation slide generation by llms with a multi-staged end-to-end approach. *arXiv preprint arXiv:2406.06556*, 2024.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

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.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2304.05128.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating LLMs by human preference. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2403.04132.

William S. Cleveland and Robert McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. LayoutGPT: Compositional visual planning and generation with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. URL https://arxiv.org/abs/2305.15393.

Tsu-Jui Fu, William Yang Wang, Daniel McDuff, and Yale Song. DOC2PPT: Automatic presentation slides generation from scientific documents. In *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, et al. Autopresent: Designing structured visuals from scratch. In *CVPR*, 2025.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing. *ICLR*, 2024.

Yiduo Guo, Zekai Zhang, Yaobo Liang, Dongyan Zhao, and Nan Duan. PPTC benchmark: Evaluating large language models for PowerPoint task completion. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 8682–8701. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.514. URL https://aclanthology.org/2024.findings-acl.514/.

Zhengkang Guo, Wenhao Liu, Mingchen Xie, Jingwen Xu, Zisu Huang, Muzhao Tian, Jianhan Xu, Yuanzhe Shen, Qi Qian, Muling Wu, Xiaohua Wang, Changze Lv, He-Da Wang, Hu Yao, Xiaoqing Zheng, and Xuanjing Huang. Recast: Expanding the boundaries of llms' complex instruction following with multi-constraint data. *arXiv preprint arXiv:2505.19030*, 2025.

Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S. Davis, Vijay Mahadevan, and Abhinav Shrivastava. LayoutTransformer: Layout generation and completion with self-attention. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. URL https://arxiv.org/abs/2006.14615.

Sirui Hong, Mingchen Zhuge, Jiaqi Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations (ICLR)*, 2024a. URL https://arxiv.org/abs/2308.00352.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. CogAgent: A visual language model for GUI agents. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b. URL https://arxiv.org/abs/2312.08914.

Dong Huang, Jie M. Zhang, Michael Luck, Qingwen Bu, Yuhao Qing, and Heming Cui. AgentCoder: Multi-agent-based code generation with iterative testing and optimisation. *arXiv preprint arXiv:2312.13010*, 2024a. URL https://arxiv.org/abs/2312.13010.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *ICLR*, 2024b.

Naoto Inoue, Kotaro Kikuchi, Edgar Simo-Serra, Mayu Otani, and Kota Yamaguchi. LayoutDM: Discrete diffusion model for controllable layout generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. URL https://arxiv.org/abs/2303.08137.

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.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.06770.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. URL https://arxiv.org/abs/2405.01535.

Stephen M. Kosslyn. *Clear and to the Point: 8 Psychological Principles for Compelling PowerPoint Presentations*. Oxford University Press, 2007.

Klaus Krippendorff. *Content Analysis: An Introduction to Its Methodology*. Sage Publications, 2 edition, 2004.

Hugo Laurençon, Léo Tronchon, and Victor Sanh. Unlocking the conversion of web screenshots into HTML code with the WebSight dataset. *arXiv preprint arXiv:2403.09029*, 2024. URL https://arxiv.org/abs/2403.09029.

Yan Li, Zezi Zeng, Ziwei Zhou, Xin Gao, Muzhao Tian, Yifan Yang, Mingxi Cheng, Qi Dai, Yuqing Yang, Lili Qiu, Zhendong Wang, Zhengyuan Yang, Lijuan Wang, Ji Li, and Chong Luo. Bizgeneval: A systematic benchmark for commercial visual content generation. *arXiv preprint arXiv:2603.25732*, 2026.

Xin Liang, Xiang Zhang, Yiwei Xu, Siqi Sun, and Chenyu You. Slidegen: Collaborative multimodal agents for scientific slide generation. *arXiv preprint arXiv:2512.04529*, 2025.

William Lidwell, Kritina Holden, and Jill Butler. *Universal Principles of Design, Revised and Updated*. Rockport Publishers, 2010.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *ICLR*, 2024.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *ICLR*, 2024.

Changze Lv, Jie Zhou, Wentao Zhao, Jingwen Xu, Zisu Huang, Muzhao Tian, Shihan Dou, Tao Gui, Le Tian, Xiao Zhou, Xiaoqing Zheng, Xuanjing Huang, and Jie Zhou. Learning query-specific rubrics from human preferences for deepresearch report generation. *arXiv preprint arXiv:2602.03619*, 2026.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, 2023.

Richard E. Mayer. *Multimedia Learning*. Cambridge University Press, 2 edition, 2009.

Richard E. Mayer (ed.). *The Cambridge Handbook of Multimedia Learning*. Cambridge University Press, 2 edition, 2014.

Tarik Can Ozden, Sachidanand VS, Furkan Horoz, Ozgur Kara, Junho Kim, and James Matthew Rehg. Narrative-driven paper-to-slide generation via arcdeck. *arXiv preprint arXiv:2604.11969*, 2026.

Yiming Pan, Chengwei Hu, Xuancheng Huang, Can Huang, Mingming Zhao, Yuean Bi, Xiaohan Zhang, Aohan Zeng, and Linmei Hu. Aeslides: Incentivizing aesthetic layout in llm-based slide generation via verifiable rewards. *arXiv preprint arXiv:2604.22840*, 2026.

Qi Qian, Chengsong Huang, Jingwen Xu, Changze Lv, Muling Wu, Wenhao Liu, Xiaohua Wang, Zhenghua Wang, Zisu Huang, Muzhao Tian, Jianhan Xu, Kun Hu, He-Da Wang, Yao Hu, Xuanjing Huang, and Xiaoqing Zheng. Benchmark^2: Systematic evaluation of llm benchmarks. *arXiv preprint arXiv:2601.03986*, 2026.

Garr Reynolds. *Presentation Zen: Simple Ideas on Presentation Design and Delivery*. New Riders, 2008.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.

Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. Design2Code: Benchmarking multimodal code generation for automated front-end engineering. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2025. URL https://arxiv.org/abs/2403.03163.

Edward Sun, Yufang Hou, Dakuo Wang, Yunfeng Zhang, and Nancy X. R. Wang. D2s: Document-to-slide generation via query-based text summarization. In *NAACL*, 2021.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. SlideVQA: A dataset for document visual question answering on multiple images. In *AAAI Conference on Artificial Intelligence*, 2023. URL https://arxiv.org/abs/2301.04883.

Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research (TMLR)*, 2024a. URL https://arxiv.org/abs/2305.16291.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024b. URL https://arxiv.org/abs/2312.08935.

Xiaohua Wang, Muzhao Tian, Yuqi Zeng, Zisu Huang, Jiakang Yuan, Bowen Chen, Jingwen Xu, Mingbo Zhou, Wenhao Liu, Muling Wu, et al. Reward hacking in the era of large models: Mechanisms, emergent misalignment, challenges. *arXiv preprint arXiv:2604.13602*, 2026.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, et al. OpenHands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2407.16741.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2201.11903.

Max Wertheimer. Laws of organization in perceptual forms. In Willis D. Ellis (ed.), *A Source Book of Gestalt Psychology*, pp. 71–88. Routledge & Kegan Paul, 1938.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. In *Conference on Language Modeling (COLM)*, 2024. URL https://arxiv.org/abs/2308.08155.

Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. LLaVA-Critic: Learning to evaluate multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2410.02712.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. Swe-agent: Agent-computer interfaces enable automated software engineering. *NeurIPS*, 2024.

Y Yang, W Li, H Ren, Z Lu, K Wang, Z Huang, et al. Slidesgen-bench: Evaluating slides generation via computational and quantitative metrics. *arXiv preprint arXiv:2601.09487*, 2026.

Yuheng Yang, Wenjia Jiang, Yang Wang, Yi Song, Yiwei Wang, and Chi Zhang. Auto-slides: An interactive multi-agent system for creating and customizing research presentations, 2025. URL https://arxiv.org/abs/2509.11062.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a. URL https://arxiv.org/abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b. URL https://arxiv.org/abs/2210.03629.

Wenzheng Zeng, Mingyu Ouyang, Langyuan Cui, and Hwee Tou Ng. Slidetailor: Personalized presentation slide generation for scientific papers. *arXiv preprint arXiv:2512.20292*, 2025. AAAI 2026.

Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. GPT-4V(ision) is a generalist web agent, if grounded. In *International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2401.01614.

Hao Zheng, Xinyan Guan, Hao Kong, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Ben He, Xianpei Han, and Le Sun. Pptagent: Generating and evaluating presentations beyond text-to-slides. In *EMNLP*, 2025. URL https://arxiv.org/abs/2501.03936.

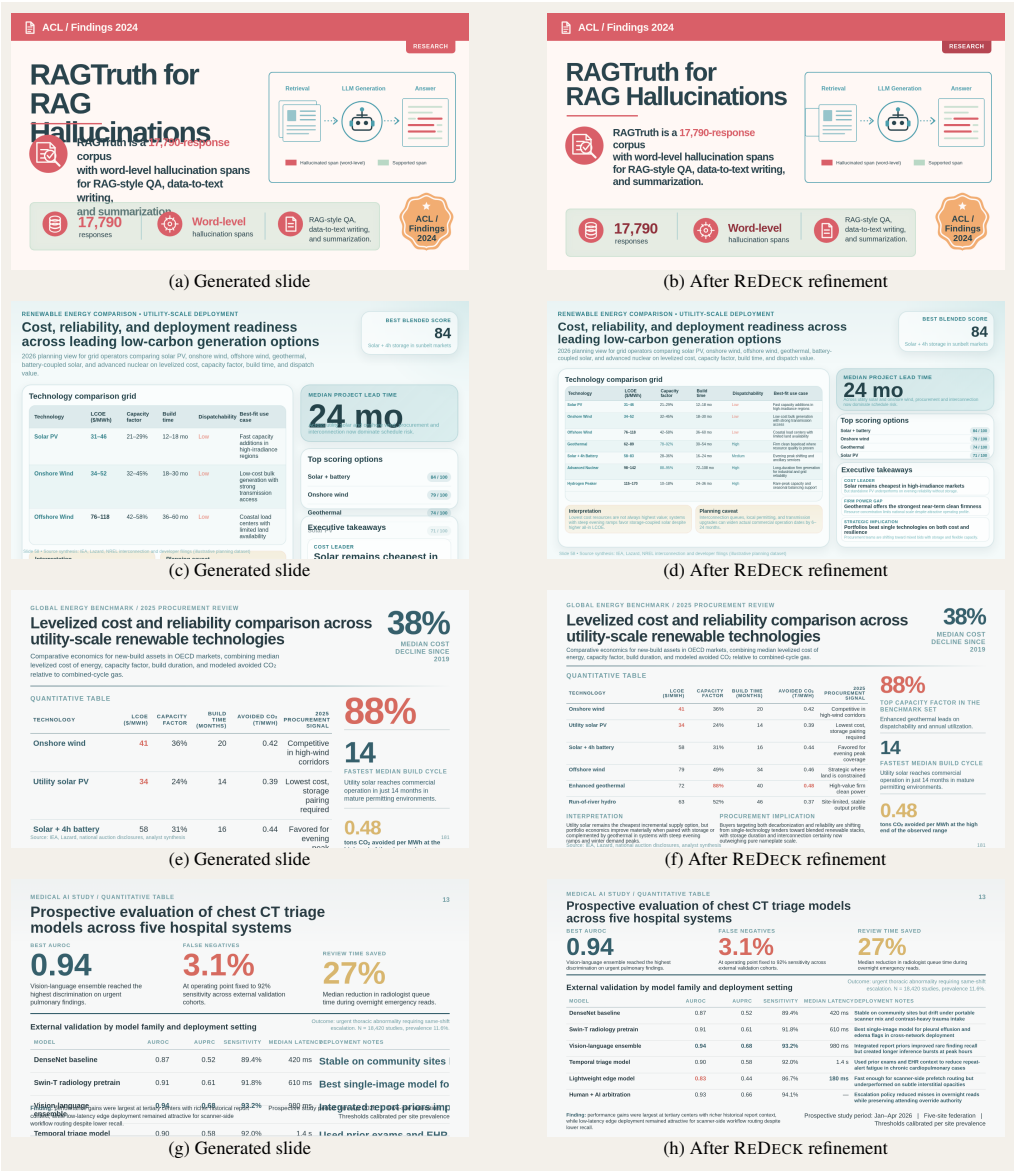
Hao Zheng, Guozhao Mo, Xinru Yan, Qianhao Yuan, Wenkai Zhang, Xuanang Chen, Yaojie Lu, Hongyu Lin, Xianpei Han, and Le Sun. Deeppresenter: Environment-grounded reflection for agentic presentation generation. *arXiv preprint arXiv:2602.22839*, 2026. ACL 2026.

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 (NeurIPS) Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2306.05685.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Yonatan Bisk, Daniel Fried, Uri Alon, et al. Webarena: A realistic web environment for building autonomous agents. *ICLR*, 2024.

A QUALITATIVE REFINEMENT EXAMPLES

Figs. 3–4 show six generated slides with naturally occurring layout defects before and after REDECK refinement. They illustrate repair behavior; Sec. 4.2 provides the quantitative evidence.



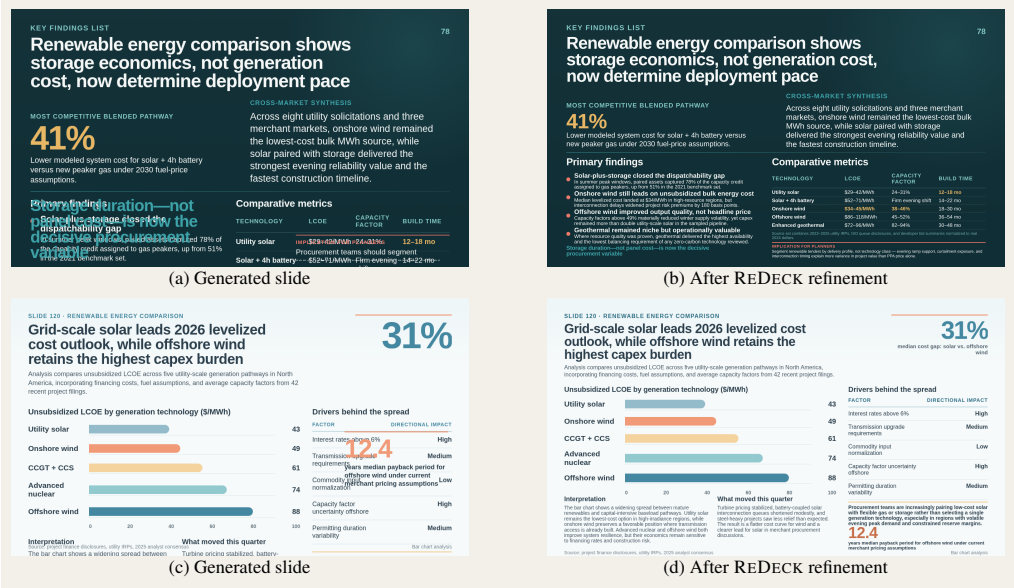
(a) Generated slide (b) After REDECK refinement

(c) Generated slide (d) After REDECK refinement

(e) Generated slide (f) After REDECK refinement

(g) Generated slide (h) After REDECK refinement

Figure 3: **Qualitative examples of REDECK refinement on naturally occurring defects.** REDECK resolves title collisions (a–b), card overflow (c–d), clipped supporting content (e–f), and table overflow (g–h).



(a) Generated slide (b) After REDECK refinement

(c) Generated slide (d) After REDECK refinement

Figure 4: **Additional qualitative examples on naturally occurring defects.** REDECK separates a findings list from the adjacent table (a–b) and removes an overlaid metric while restoring lower-page content (c–d).

B INITIAL-DRAFT PIPELINE AND RENDERING ENVIRONMENT

This appendix documents two implementation layers that sit *outside* the REDECK refinement loop: (i) the bootstrap pipeline that produces its session-start deck d_0, and (ii) the rendering environment that turns editable deck state into rendered geometry r_t and diagnostics C_t. The refinement loop itself — the compositional action space, Step-Level Render Feedback, the Adaptive Deck Critic, and the submission gate ρ — is specified in §3. Every REDECK ablation in Sec. 4.3 starts from the same d_0 and uses the same renderer and extractor. Published baselines use their native initial-generation pipelines in Tab. 1; App. O reports the corresponding initial-deck attribution analysis.

Bootstrap: producing the session-start deck d_0. Given a source paper, the bootstrap runs four ordered steps and emits a complete HTML/CSS deck that becomes d_0 in the notation of §3.1. *(i) Source parsing* converts the paper (PDF or LaTeX source) into structured sections via layout analysis combined with an LLM section classifier; tables, figures, and equations are extracted separately and stored as referenceable entities with their captions and source locations. *(ii) Blueprint planning* issues a single LLM call that reads the parsed structure and emits a JSON *deck blueprint* that assigns each slide a role (title, outline, content, figure, comparison, conclusion) and selects the source spans that populate it. *(iii) Layout selection* maps each blueprint entry to one template from a library of 15 base layouts (single-column, two-column, figure-dominant, table-dominant, etc.), conditioned on the slide role, content density, and figure count from the blueprint. *(iv) Code generation* issues one code-generation LLM call per slide that consumes the blueprint entry, layout template, and source spans; per-slide independence makes this step embarrassingly parallel across the deck. The result is the deck source d_0 that the REDECK family of variants in Sec. 4.3 shares as its starting state. Anything labelled "no refinement" or "T_0" in the experiments refers to this d_0 rendered without any loop applied on top. Published baselines (SLIDEGEN, SLIDETAILOR, DEEPPRESENTER) are run on their released code under the matched per-task LLM-call-count cap and use their own native initial-generation pipeline.

Rendering environment: producing r_t and C_t. Generated HTML slides are rendered to PNG via a headless browser (Playwright/Chromium at 1280×720, the same canvas as the case study in App. C). A DOM-based spatial extractor traverses the rendered DOM tree and computes bounding boxes, an overlap matrix, overflow measurements, and off-canvas indicators; together with the compiler diagnostics C_t collected during rendering, these are the rendered-side components of the hidden state $s_t = (d_t, r_t, C_t)$ defined in §3.1. Step-Level Render Feedback (§3.2) reads from this extractor

to construct the observation $o_t = (\Delta_{\text{viol}}(d_t, d_0),\ A_t)$ delivered after every atomic action, and the Adaptive Deck Critic (§3.3) reads from the same extractor (alongside the rendered PNGs) when its deterministic-geometry probes update L_{t+1}. For final SCR reporting, all systems are rendered to PNG and evaluated by the same SpatialCheck VLM judge (App. F); the DOM extractor is used for REDECK's internal step-level feedback rather than as the cross-system SCR scorer.

C CASE STUDY: STEP-LEVEL RENDER FEEDBACK PREVENTS CASCADING LAYOUT REGRESSIONS

Fig. 5 traces two editing sequences from the same starting slide: one without step-level feedback and one with REDECK. This simple case study shows how feedback timing changes the repair trajectory; the controlled experiments in Sec. 4.3, rather than this single example, support the quantitative comparisons. The initial state s_0 contains three observed issues, comprising one visual-style issue and two geometric violations: i_1 a low-contrast title (foreground / background lift below readability threshold), i_2 a bullet column whose right edge overlaps the chart container, and i_3 a takeaway callout positioned with `left` $= 940$, `width` $= 400$ on a 1280-wide canvas, so its right edge lies 60 px off-canvas. The left column shows the trajectory without step-level feedback: the agent receives rendered critique only at turn boundaries, so multiple edits can accumulate before any visual feedback is observed. The right column shows REDECK with step-level render feedback on the same state, where each atomic edit is rendered and checked before the trajectory advances.

Without step-level render feedback (left). The baseline receives a turn-boundary verdict (BULLET/CHART OVERLAP; CALLOUT LOOKS LOST AT THE BOTTOM) and applies both suggestions in a single batch as e'_1: narrows the bullet column from 800 to 580 px (well intentioned, addresses i_2) and simultaneously raises the callout from $y = 540$ to $y = 300$ to "increase visibility." Without per-step rendering it does not see that the new callout position now sits on top of the chart at $x = 940-1240$, $y = 300-480$. The next critic verdict (CALLOUT OVERLAPS CHART) drives e'_2: enlarge the chart to 720 px wide so the callout is no longer in front of it; the new chart right edge is at $x = 1340$, off-canvas, but the agent has no way to see this. A third verdict (LOWER-LEFT FEELS EMPTY, ADD METRICS) drives e'_3: a three-card metric strip is inserted at `top` $= 470$, colliding with the bullet text that already extends to $y \approx 480$. After these three edits, the trajectory contains five issues rather than three. These counts include the low-contrast title and are not measurements of the gate's hard-violation count.

With step-level render feedback (right). Within turn 0 REDECK issues four atomic edits. e_1 narrows the bullet column to 580 px alone (no batched callout move); the agent observes the render report and keeps the edit ($s_0 \to s_1$). The feedback channel itself does not accept or reject edits. e_2 then attempts to widen the chart to 720 px to use the freed horizontal room; the new chart right edge lands at $x = 1340$, breaching the canvas, and step-level render feedback reports the regression immediately; the agent chooses `rollback`, returning the deck to s_1 before the trajectory continues (the inset on the right column shows the rolled-back attempt struck through with a red cross). e_3 takes a different shape: the takeaway becomes a full-width bottom band (`left` $= 40$, `width` $= 1200$), resolving i_3. e_4 darkens the title text from `rgb`$(180, 188, 195)$ to the Slate Modern primary `rgb`$(38, 50, 56)$, resolving i_1. At the turn-0 boundary the deck critic confirms B_{geom} as clean and raises a new $A_{\text{narrative}}$ finding: the chart alone is hard to parse in three seconds. Turn 1 contains a single edit: e_5 inserts a three-card metric strip below the chart that surfaces the key numbers. The final rendered state retains zero hard geometric issues after the narrative repair.

What the case study shows. The two trajectories show three behaviours in this case, without estimating their frequency: (i) step-level render feedback catches the off-canvas chart (e_2) one edit after it appears, so detection costs one rollback rather than a multi-edit pile-up; (ii) rollback returns to the preceding retained state, which may still contain other violations, so the agent can try a different edit shape (e_3); (iii) the deck critic can assess signals beyond the geometry channel — it raises a turn-boundary narrative finding (e_5), and the next step-level check confirms that the narrative repair did not silently regress geometry.

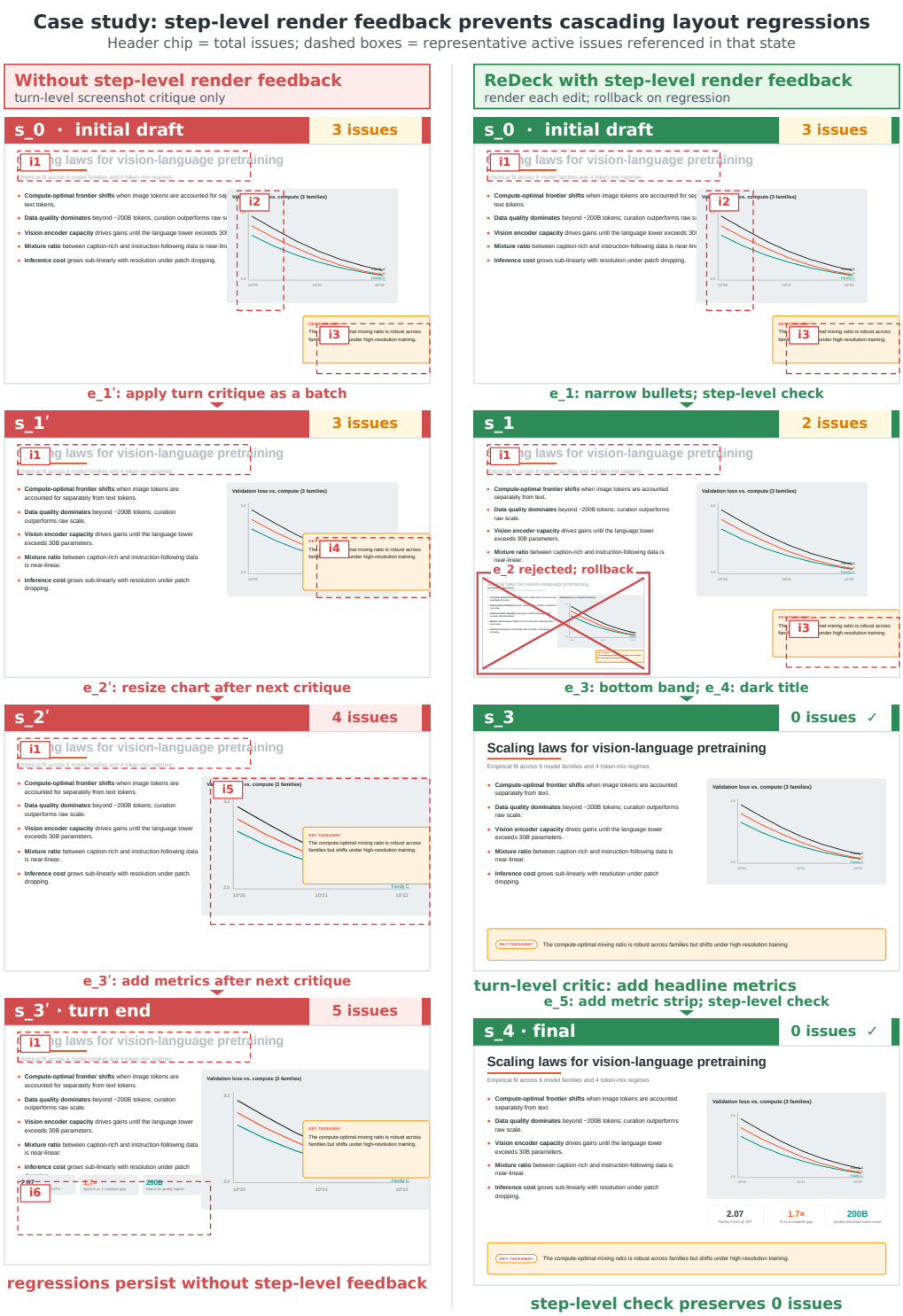


Figure 5: **Case study: step-level render feedback prevents cascading layout regressions.** Both columns start from the same draft with two geometric issues and a low-contrast title; displayed counts include both types, not just gate violations. **Left**: without step-level feedback, well-intentioned edits silently introduce new issues (count grows $3 \rightarrow 5$). **Right**: the agent observes the off-canvas regression, chooses rollback, and tries a different edit. This recorded case explains the mechanism but does not estimate its frequency.

D DECKQUIZ HUMAN ALIGNMENT STUDY

We evaluate subjective judge alignment on 50 held-out decks per module using three blinded non-author raters. This calibration measures agreement on Aesthetics axes and DeckDesign principles, not pairwise system preference (App. E). SpatialCheck has a separate 150-slide validation in App. F.

Sampling and uncertainty. Each module supplies five axis/principle scores per deck, yielding 250 deck–axis/principle items and 750 human ratings per module. These items share decks and are not independent samples. We report ordinal Krippendorff's α (Krippendorff, 2004) for human reliability, and quadratic-weighted κ, Spearman correlation, and normalized mean absolute error (NMAE) against the human median. Raw MAEs are 0.42 on the 0–5 aesthetics scale and 0.38 on the 0–3 design scale; dividing by the scale ranges gives NMAEs of 8.4% and 12.7%. Confidence intervals resample decks and account for axis/principle structure.

Table 4: Subjective-module calibration on 50 decks per module with three non-author raters. Brackets are 95% CIs clustered by deck.

Module	Human α	VLM–human κ	Spearman ρ	NMAE
Aesthetics	0.71 [0.63, 0.78]	0.65 [0.56, 0.73]	0.72 [0.61, 0.81]	8.4%
DeckDesign	0.68 [0.59, 0.76]	0.61 [0.51, 0.70]	0.69 [0.57, 0.79]	12.7%

Agreement is imperfect, particularly for DeckDesign; these scores support using the subjective modules alongside human preference and external evaluation, not as interchangeable with human judgments. ContentQuiz answer screening is described in App. L; this subjective calibration does not validate its QA labels.

E DECK-LEVEL HUMAN PAIRWISE STUDY

This appendix details the forced-choice deck-level study referenced in Sec. 4.5. Its purpose is external validation against the strongest baseline, distinct from the per-item judge calibration in App. D.

Items and pairing. A 45-task subset is drawn from the 100-task GPT-5.4 split. For each task we form one forced-choice pair (Bradley & Terry, 1952; Chiang et al., 2024), REDECK versus DeepPresenter, yielding 270 individual judgments across two dimensions and three annotators.

Annotators and blinding. Three annotators with prior slide-design or HCI experience, none of whom contributed to the probe library, critic prompt, or DECKQUIZ rubric design. Decks are rendered to PNG at the same 1280×720 canvas used by the loop (App. B) and then up-sampled to a fixed-DPI PDF for printing; file names are stripped of system identity, left/right order is randomised independently per pair, and the same pair is never shown twice in a row to the same annotator.

Task and dimensions. Each annotator answers two forced-choice questions per pair: (D1) *which deck better presents the source paper overall* (presentation quality), and (D2) *which deck is more faithful to the source paper* (faithfulness). A third "no preference" option is allowed but counted as 0.5 for each side; cases with $\geq 2/3$ "no preference" are reported separately and excluded from the ranking.

Statistics. We aggregate the three ratings by task-level majority vote and report REDECK's preference rate with Wilson 95% CIs (Wilson, 1927). No task has a majority "no preference" outcome.

Table 5: Blinded human preference for REDECK over DeepPresenter on 45 tasks (3 annotators per task, majority vote).

Dimension	W/L/T	Preference	Wilson 95% CI
Overall presentation quality	33/12/0	0.73	[0.59, 0.84]
Source faithfulness	31/14/0	0.69	[0.54, 0.80]

F SPATIALCHECK VLM-JUDGE VALIDATION

This appendix calibrates the vision-language judge that produces the SpatialCheck *clean-rate* (SCR) reported in Tab. 1, so that SCR differences between systems can be attributed to deck quality rather than judge noise. Unlike the DOM-based extractor used inside REDECK's step-level render feedback (App. B), which operates only on HTML/CSS source, the SCR judge consumes the rendered PNG of each slide and therefore applies uniformly to PPTX, PDF, and HTML decks across all comparison systems.

Judge. A GPT-5.4 vision model receives one rendered slide at a time together with a fixed rubric enumerating four hard-violation families and returns a strict-JSON list of per-family violations; this follows the rendered-screenshot vision-judge paradigm developed for multimodal evaluation (Xiong et al., 2025). The four families are `S1 text_clipping` (visible text cut off by a container or canvas edge), `S2 element_overlap` (two or more readable elements occlude one another), `S3 out_of_bounds` (an element lies partially or wholly outside the slide canvas), and `S4 image_clipping` (an image is cropped by its bounding container so that semantic content is lost). A slide counts as *clean* iff its violation list is empty across all four families; SCR is the fraction of clean slides over the deck.

Sample. We sample 150 rendered slides from the shared 100-paper task split, stratified across systems, agent backbones, and the four hard-violation families.

Reference labels. Two non-author raters independently label each slide and adjudicate disagreements before seeing the VLM predictions.

Metrics. We report inter-rater Cohen's κ and VLM sensitivity, specificity, and F_1 against the adjudicated reference. Dirty/violation is the positive class.

Table 6: SpatialCheck validation against an adjudicated non-author reference on 150 slides.

Target	Inter-rater κ	Sensitivity	Specificity	VLM F_1
Clean/dirty	0.78 [0.67, 0.87]	0.86	0.92	0.87 [0.80, 0.93]
Text clipping	0.74 [0.62, 0.84]	0.84	0.90	0.85 [0.79, 0.91]
Element overlap	0.68 [0.56, 0.79]	0.78	0.88	0.80 [0.72, 0.87]
Out of bounds	0.82 [0.73, 0.91]	0.89	0.93	0.89 [0.84, 0.94]
Image clipping	0.71 [0.59, 0.81]	0.80	0.89	0.81 [0.74, 0.88]

The primary clean/dirty balanced accuracy is 0.89. As a deterministic consistency check, we also score all six HTML variants with the DOM extractor used by the step-level channel. Across 100 tasks $\times$ 3 seeds $\times$ 6 variants, DOM-SCR preserves the full VLM-SCR ordering, with 92% slide-level agreement ($\kappa = 0.83$ [0.79, 0.86], MCC $= 0.84$ [0.80, 0.87]), a deck-level bias of -0.8 points $[-1.3, -0.3]$, and MAE 1.4 points [1.0, 1.8]. This is a second implementation of the same violation taxonomy, not an independent task definition.

Table 7: VLM-SCR and deterministic DOM-SCR on the same HTML variants. Differences are DOM minus VLM in percentage points; CIs for agreement statistics resample source tasks.

Variant	VLM-SCR	DOM-SCR	Difference
T_0	64.1	63.2	-0.9
NL critique	68.2	67.0	-1.2
Screenshot critique	78.4	77.5	-0.9
Turn-level critic only	60.8	59.9	-0.9
Action-level render only	80.2	79.8	-0.4
REDECK	91.5	90.8	-0.7

G DATA OVERLAP AND EVALUATION COUPLING

Title/DOI matching found no source-paper overlap between the 100-paper DeckQuiz pool and PresentBench. The probe library was frozen before quiz construction and did not use evaluation items.

This separates item reuse from construct overlap: both probe design and evaluation still concern fidelity, geometry, and design. Matching identifiers does not establish absence from foundation-model pretraining. External evaluations in App. H and alternative judges in App. J.1 address complementary aspects of evaluator dependence.

H External Evaluation

We evaluate the same five systems using DECKBench (Jang et al., 2026), SlidesGen-Bench (Yang et al., 2026), and AeSlides (Pan et al., 2026). The first two change both data and evaluator; AeSlides changes the evaluator on the same 100 DeckQuiz tasks. Three seeds are averaged within the native unit before aggregation, and confidence intervals bootstrap that unit. The five-comparison Holm family comprises ReDeck–DeepPresenter contrasts on LayoutQ, DeckFaith, DeckFid, QuizBank, and AeSlides E.W.; other metrics, including computational aesthetics, are secondary.

Completion and scope. DECKBench quality scores use the common completed set of 281 of 294 paper–slide pairs after PDF export; native failure rates use all $294 \times 3 = 882$ attempted seed-runs per system. SlidesGen-Bench uses the common completed set of 181 of 187 topics. Restricting comparisons to common completions ensures pairing but conditions on successful execution; the quality means should therefore be read alongside failures, not as unconditional performance on all attempts.

Table 8: External evaluation on the common completed sets. LQ, DFaith, and DFid denote DECK-Bench LayoutQ, DeckFaith, and DeckFid; Quiz and Aes. are SlidesGen-Bench metrics.

System	LQ ↑	DFaith ↑	DFid ↑	Quiz (%) ↑	Aes. ↑
SlideGen	0.932	0.643	0.556	75.8	20.8
SlideTailor	0.948	0.631	0.548	73.9	22.6
DeepPresenter	0.936	0.658	0.566	76.4	21.7
ReDeck T_0	0.941	0.669	0.575	77.2	20.1
ReDeck	**0.968**	**0.681**	**0.588**	**79.0**	**24.9**

Relative to DeepPresenter, ReDeck gains 0.032 [0.018, 0.046] on LQ, 0.023 [0.010, 0.036] on DFaith, 0.022 [0.008, 0.036] on DFid, 2.6 percentage points [0.8, 4.4] on Quiz, and 3.2 points [1.5, 4.9] on Aes. Quiz measures answerability from extracted slide content rather than complete source fidelity; the results therefore triangulate, rather than duplicate, the four DeckQuiz modules.

All evaluated DECKBench reference-free metrics. Faith measures slide-to-paper grounding; Deck-Faith uses one-to-one slide–paper matching; DeckFid measures paper coverage; LayoutQ is the in-canvas element fraction. PPL and DeckPPL are the protocol's slide- and deck-level perplexities. ReDeck does not minimize either perplexity, so improvements are not uniform across the protocol. Reference-based TextSim, FigAlign, DTW, and TransSim are not evaluated because this comparison targets valid alternative presentations rather than similarity to one reference deck.

Table 9: All DECKBench Task 1 reference-free metrics. Quality: 281 common pairs. Failure rate: 882 attempted seed-runs per system.

System	PPL ↓	Faith ↑	LayoutQ ↑	DeckPPL ↓	DeckFaith ↑	DeckFid ↑	Fail% ↓
SlideGen	176.4	.507	.932	31.6	.643	.556	1.4
SlideTailor	148.2	.493	.948	28.9	.631	.548	1.1
DeepPresenter	152.7	.526	.936	29.7	.658	.566	0.7
ReDeck T_0	164.6	.532	.941	31.4	.669	.575	0.9
ReDeck	158.9	.544	.968	30.5	.681	.588	0.0

SlidesGen-Bench protocol. A fixed VLM extracts content from each rendered deck, and a fixed text LLM answers the released question bank from that extraction. The computational-aesthetics column in Tab. 8 is the published aggregate, not DeckQuiz Aesthetics. Answering models may infer information not explicitly present, so QuizBank is an answerability check rather than a complete fidelity audit.

AeSlides checks. The image-based E.W. metric and native render/export failure rate apply to all formats. A.R. compliance at the 1% and 5% thresholds and V.I. require HTML runtime geometry; these are reported only for the HTML subset. We retain the metric names and directions from the external protocol. E.C. is omitted because the complete evaluator is unavailable.

Table 10: AeSlides checks on 100 tasks. Panel (a) supports all native formats; panel (b) requires HTML geometry. A.R. entries are percentages at the two thresholds.

(a) Format-uniform metrics

System	Render/export fail% ↓	E.W. ↓
SlideGen	1.2	.058
SlideTailor	0.6	.050
DeepPresenter	0.8	.054
REDECK T_0	0.9	.060
REDECK	0.0	.043

(b) HTML-only metrics

System	A.R. 1%/5% ↑	V.I. ↓
DeepPresenter	57/74	1.34
REDECK T_0	49/68	1.46
REDECK	76/88	1.12

Against DeepPresenter, REDECK reduces E.W. by 0.011 [0.004, 0.018] across formats. Within the HTML subset it raises A.R. compliance by 19/14 percentage points and reduces V.I. by 0.22 [0.10, 0.34]. The latter is secondary evidence. Different implementations and datasets reduce evaluation coupling, but do not make the underlying quality constructs unrelated.

I TRANSFER TO PPTX

For PPTX, REDECK replaces browser DOM queries with `python-pptx` shape geometry and renders each state through LibreOffice. The action loop, critic, rollback, and submission gate are unchanged.

Table 11: PPTX refinement on GPT-5.4 (100 tasks × 3 seeds).

PPTX state	Fid. ↑	SCR ↑	Aes. ↑	Des. ↑
PPTX T_0	78.0	58.0	2.80	60.0
PPTX + REDECK	83.0	82.0	3.30	66.0

The gains are $+5.0/+24.0/+0.50/+6.0$ on FID./SCR/AES./DES.; the SCR gain has a 95% CI of [19.1, 28.9]. This supports transfer to a second editable geometry/rendering stack, but does not establish support for PDF/OCR-only editing, Keynote, animations, masking, or font-fallback behavior.

J JUDGE CONFIGURATION

The main tables use a fixed **GPT-5.4** evaluator regardless of the agent backbone, with vision enabled for SCR, Aesthetics, and DeckDesign. Fixing the evaluator makes rows comparable under one scoring configuration, but does not remove same-family self-preference when the generator is also GPT-5.4. We therefore combine human calibration with the alternative-evaluator analysis below. Judge bias is documented in prior LLM-as-a-judge and rubric-based evaluation work (Zheng et al., 2023; Chiang et al., 2024; Kim et al., 2024; Xiong et al., 2025). The ContentQuiz QA grader for FID. uses the same backbone in text-only mode against the held-out QA bank in App. L. Evaluation calls use deterministic decoding (temperature 0, top-p 1, fixed seed where the API exposes one) and a strict structured-JSON output schema; parse failures trigger a single retry, after which the call is recorded as missing and excluded from the corresponding cell mean (less than 0.3% of calls in the

main run). Agent calls, in contrast, keep each system's released defaults (sampling temperature, top-p, and tool-use settings as shipped) so that we benchmark each baseline at its publicly recommended operating point rather than a re-tuned variant. The judge prompt templates are released with the codebase at `app/prompts/`.

J.1 Sensitivity to the Evaluator Backbone

We rescore the same GPT-5.4-generated decks with GPT-5.4, Claude-4.6, and Gemini-3.1 evaluators, holding the question bank, rubrics, prompts, and decoding settings fixed. This varies the judge, not the generator. ContentQuiz uses text-only reader mode; the other modules use rendered images. Means aggregate 100 tasks and three seeds.

Table 12: Evaluator sensitivity on fixed GPT-5.4 outputs. ReDeck has the highest mean in each module for all three evaluators, but subjective score differences are not uniformly significant.

Evaluator	System	Fid. ↑	SCR ↑	Aes. ↑	Des. ↑
GPT-5.4	SlideGen	65.3	83.4	3.08	50.2
	SlideTailor	60.2	88.2	2.90	56.4
	DeepPresenter	76.8	66.8	3.47	55.1
	ReDeck	88.6	91.5	3.64	71.2
Claude-4.6	SlideGen	61.8	78.6	2.71	45.6
	SlideTailor	62.4	82.9	3.04	58.9
	DeepPresenter	73.9	71.4	3.18	60.7
	ReDeck	84.7	86.8	3.31	66.8
Gemini-3.1	SlideGen	68.7	86.1	3.34	54.3
	SlideTailor	64.9	85.4	3.08	61.5
	DeepPresenter	78.1	73.8	3.56	57.8
	ReDeck	86.2	88.1	3.71	74.6

ReDeck remains first in all eight alternative-evaluator module comparisons. The complete four-system ordering matches the GPT-5.4 ordering in four of eight cases; each remaining case has one non-leading inversion (Kendall's $\tau = 0.67$). Seven of eight task-paired ReDeck–DeepPresenter CIs exclude zero. The exception is Claude Aesthetics, $+0.13$ $[-0.03, 0.29]$, so stable top means do not establish evaluator-invariant superiority.

After removing system means across 400 system–task pairs, Spearman correlations with the GPT-5.4 evaluator are 0.82/0.77 for fidelity, 0.87/0.80 for SCR, 0.64/0.71 for aesthetics, and 0.69/0.75 for design (Claude/Gemini). Subjective calibration and lower-ranked systems remain judge-sensitive.

K Pilot Audit, Taxonomy Mapping, and Probe Coverage

This appendix establishes that the six probe families in §3.3 (A narrative, B_{geom} layout, B_{visual} visual style, C completeness, D correctness, E source fidelity) are derived from rather than imposed on the failure distribution, and documents their provenance from external benchmarks.

External provenance. Tab. 13 maps each ReDeck probe family to the external failure dimensions from which it is derived.

Sample. We hand-label 100 refinement turns drawn at random from preliminary ReDeck runs collected before the probe library used in the main evaluation was finalised. Each turn contributes one to several discrete failures; the unit of analysis is the *failure*, of which the 100 turns yield 264 instances.

Schema and inter-rater agreement. Two authors independently assign each failure to one of the six families or to an *other* bucket using a written codebook released with the supplementary code. Disagreements are resolved by adjudication after both passes are complete to avoid leakage between annotators. We report Cohen's κ for classification among the six probe families as inter-rater agreement. The residual *other* bucket is not a seventh class in this calculation; it is reported separately to assess taxonomy coverage.

Table 13: External failure taxonomies projected onto the REDECK probe families.

REDECK family	External sources and mapped dimensions
Narrative	Thesis alignment, flow, and title-content mismatch failures.
Visual & Layout	Public render constraints (aspect-ratio, element collision) and softer design signals (whitespace, alignment, density) from AESLIDES.
Completeness	Missing-content and content-completeness failures from SLIDESGEN-BENCH, PRESENTBENCH, and DEEPPRESENTER.
Correctness	Entity, number, date, and table-value mismatch failures from SLIDESGEN-BENCH and PRESENTBENCH.
Source Fidelity	Unsupported visible claims and weak evidence traceability from PRESENTBENCH Fidelity and DEEPPRESENTER.

Coverage criterion. The taxonomy is accepted if (i) the residual *other* rate is at most 5% of failures and (ii) Cohen's $\kappa \geq 0.6$ (Cohen, 1960). Failure to meet either criterion forces taxonomy revision before the probe library is committed.

Table 14: Pilot audit: distribution of 264 hand-labelled failures over the six probe families and the residual *other* bucket, with two-author Cohen's κ.

Family	Count	Share
A narrative	38	14.4%
B_{geom} layout	72	27.3%
B_{visual} visual style	41	15.5%
C completeness	52	19.7%
D correctness	31	11.7%
E source fidelity	17	6.4%
Other (residual)	13	4.9%
Two-author Cohen's κ		0.78

Fig. 6 renders the same distribution as a donut chart with the residual *other* slice exploded for emphasis; the centre of the donut carries the two-author κ together with the total failure count, and the acceptance criterion is annotated above the chart so that audit success or failure can be read off without referring back to the table.

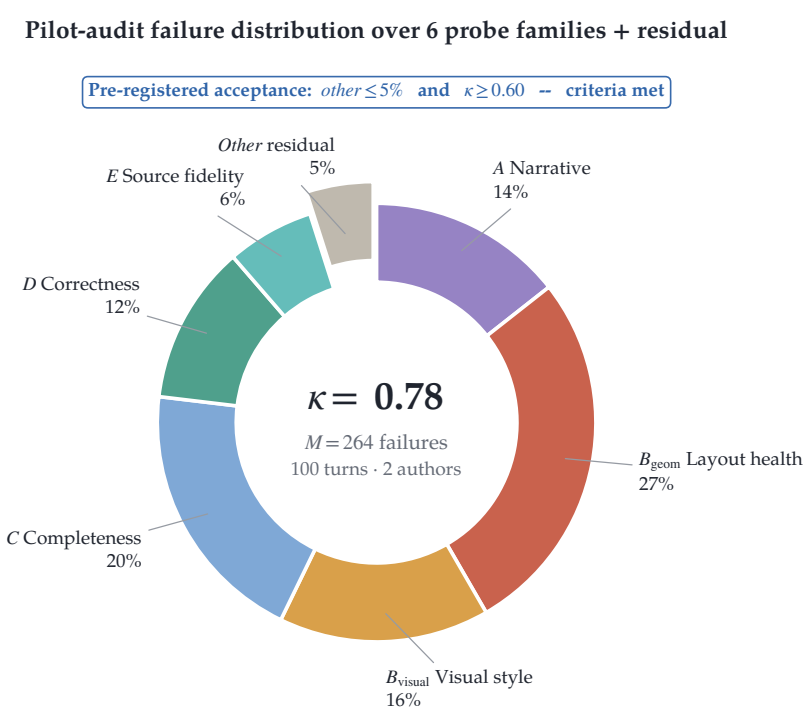


Figure 6: **Pilot-audit failure distribution over the six probe families.** Each ring slice is one probe family, sized by its share of the 264 hand-labelled failures from the 100-turn pilot; the small grey wedge labelled *Other* is the residual unclassified bucket, exploded to make its size visible against the $\leq 5\%$ ceiling. The donut centre reports the two-author Cohen's κ used as the inter-rater check; the green badge above the chart confirms that both acceptance criteria (*other* $\leq 5\%$ and $\kappa \geq 0.60$) are satisfied.

L DeckQuiz Construction and Protocol

Source pool. The 100 evaluation papers are drawn from five disciplines (CS/ML 25, Bio-Med 15, Physical-Sci 25, Econ/Finance 25, Soc-Sci/HCI 10). Across the full evaluation split, 30 papers are short ($\leq$8 pages), 40 are medium (9–15 pages), and 30 are long (>15 pages). All papers are from arXiv and are selected to have no pre-existing official presentation slides so that no system in Tab. 1 could have memorised the target deck during pre-training. The paper-grounded QA construction follows the document-VQA tradition (Tanaka et al., 2023) and the slide-from-paper benchmarking lineage (Fu et al., 2022), but localises questions to per-paper extracted facts so that the deck rather than a topic-level summary is being evaluated.

Question generation: what should a reader leave the deck knowing? ContentQuiz operationalises a reader-centric question: *after watching the deck once, can the audience answer the questions a reviewer would ask about this paper?* For each paper we instantiate this with a four-type taxonomy that mirrors the canonical reviewer rubric: *Contribution* questions probe the paper's main claims and novelty; *Method* questions probe the proposed approach; *Experiment* questions probe the key results and the conditions under which they hold; *Limitation* questions probe the caveats and scope restrictions. For each type the generator is conditioned on the paper alone (not the deck) and asked to write a question whose answer must be present in a faithful deck; this decouples question difficulty from the system being evaluated and prevents FID. from rewarding decks that happen to share a stylistic prior with the judge.

Span-extraction templates. Each question is paired with the source span that contains the answer. Spans are extracted by a template-guided LLM call that localises the answer to a contiguous passage of $\leq$200 tokens in the source paper, supporting item-level grounding and audit of the generated question and gold answer.

Why quiz, not checklist. PRESENTBENCH (Chen et al., 2026) and the DEEPPRESENTER eval suite (Zheng et al., 2026) score content fidelity by a per-deck *checklist* of binary "is property X present?" rubric items. A checklist conflates two failure modes that an iterative refinement loop must distinguish: (a) the deck omits a substantive fact the audience needs, and (b) the deck phrases a fact in a form the rubric did not anticipate. A reader-centric quiz collapses (b) because the question is the canonical phrasing and the four answer choices fix the scoring surface, while (a) remains diagnosable via per-type accuracy. MC distractors are drawn from related papers in the same discipline (not generic foils) so that a deck which talks about the right *topic* but omits the right *paper-specific* content cannot pass by topic match alone — a failure mode that affects every refinement system that paraphrases freely.

Protocol. For each paper, 5 questions per type are generated (20 total), each with 4 answer choices (one correct, three same-discipline distractors). Every question must be answerable from a well-made deck alone, without consulting the full paper. The judge LLM (App. J) receives deck content in text-only reader mode and selects an answer for each question; FID. is the fraction of correct answers, averaged across all four types.

Answer uniqueness and No-Context screening. A calibration pass removes questions whose correct answer can be guessed from world knowledge alone. Under the No-Context condition, the judge receives only the question and answer choices; questions answered correctly in more than half of three trials are regenerated.

Statistics. The final bank contains 2,000 questions across 100 papers (20 per paper); per-type distribution: 500 Contribution / 500 Method / 500 Experiment / 500 Limitation; mean question length 18.3 tokens, mean answer-choice length 8.7 tokens.

License. Evaluation code is released under the MIT license. Dataset artifacts are distributed according to per-item source licenses documented in a license manifest accompanying the release; for sources that do not permit redistribution, we release identifiers and reconstruction scripts only.

Comparison against existing slide benchmarks. Tab. 15 positions DECKQUIZ against existing deck-level benchmarks along the axes that matter for refinement-loop attribution: how spatial health is measured, how content fidelity is scored, how design quality is anchored, and whether the suite resolves per-loop-component (step-level vs. turn-level) signal beyond a single end-to-end deck score. The differentiating column is *per-component attribution*: most existing benchmarks score the final artifact as a whole, making it difficult to isolate the contribution of individual loop components. DECKQUIZ pairs an explicit four-family definition of spatial violations with a calibrated rendered-PNG VLM judge, paper-level QA fidelity, and a per-layer principle decomposition, supporting analysis of individual refinement components.

Table 15: Comparison of DECKQUIZ against existing deck-level evaluation suites. *Spatial*: rendered-PNG VLM judge (LLM) vs. deterministic DOM/render-based geometric check (DOM) vs. absent (no). *Content*: per-paper QA against extracted facts (QA) vs. rubric / checklist LLM scoring (rubric) vs. embedding similarity to a reference deck (sim) vs. absent (no); "+" means both modes are supported. *Aesthetic*: VLM rubric over multiple design axes (LLM) vs. low-level visual statistics or verifiable layout reward (signal) vs. absent (no). *Human anchor*: forced-choice human study calibrating the judge. *Per-component attr.*: whether the suite resolves per-loop-component signal beyond end-to-end ranking. *Trajectory*: whether the suite reports per-turn or per-iteration metrics. Numbers from each benchmark's published paper / repository; AeSlides counts individual page-level samples (†), not decks; SlidesGen-Bench counts generated outputs (‡); "—" marks dimensions not specifically curated.

Benchmark	# units	Unit type	Spatial	Content	Aesthetic	Human anchor	Trajectory	Per-comp. attr.
PRESENTBENCH (Chen et al., 2026)	238	tasks	LLM	rubric	LLM	yes	no	no
SLIDESGEN-BENCH (Yang et al., 2026)	1,683‡	outputs	DOM	QA	signal+LLM	yes	no	no
AESLIDES (Pan et al., 2026)	6,736†	pages	DOM	no	signal	yes	no	no
DECKBENCH (Jang et al., 2026)	294	pairs	DOM	sim+rubric	no	no	yes	no
PPTAGENT eval (Zheng et al., 2025)	500	decks	no	rubric	no	partial (32 decks)	no	no
DECKQUIZ (ours)	100	tasks	LLM	QA (paper-level)	LLM	yes	yes	yes (4-tier)

M DECKDESIGN PRINCIPLES AND SCORING

This appendix specifies the five principles whose aggregate is the DES. column of Tab. 1 and Tab. 3. The principles are organised into two layers that capture the *information architecture* of a slide deck—an orthogonal dimension to geometry (SpatialCheck), aesthetics (Aesthetics), and content fidelity (ContentQuiz). The theoretical grounding draws on Mayer's multimedia principles (Mayer, 2014), Kosslyn's cognitive-load guidelines (Kosslyn, 2007), Cleveland & McGill's graphical perception hierarchy (Cleveland & McGill, 1984), Tufte's data–ink ratio (Tufte, 1983), and Reynolds' presentation-Zen heuristics (Reynolds, 2008). Layer 1 (deck-level) requires seeing the full slide sequence and is judged once per deck; Layer 2 (slide-level) is judged per slide and averaged.

Layer 1: Deck-level (P1, P5). *P1 Narrative Progression.* The deck builds a coherent argumentative arc (context → gap → approach → evidence → implication), and each slide advances the narrative by introducing substantive information the audience needs at that point; content-free slides (section dividers, outlines, thank-you placeholders) break progression and are penalised. *P5 Cross-Slide Consistency.* Recurring concepts (variable names, model names, colour encodings, terminology) are referred to with consistent symbols and visual encodings throughout the deck so it reads as one integrated document rather than a patchwork of independently authored slides.

Layer 2: Slide-level (P2–P4). *P2 Slide Segmentation* (Mayer's segmenting principle): each slide conveys exactly one coherent message unit, neither overloaded (multiple topics on one slide) nor underloaded (title-only divider). *P3 Signal & Emphasis* (Mayer's signaling principle): the slide uses visual hierarchy (size, weight, colour, position) to direct attention to the key takeaway, evaluated along three aspects—(a) action-oriented assertion titles vs. generic topic labels, (b) differential body hierarchy vs. uniform template styling, and (c) explicit takeaway elements (summary bars, callout boxes). *P4 Structural Correspondence* (Cleveland & McGill, Tufte): the visual form matches the content's logical structure—tables for comparisons, charts for quantities, timelines for processes, prose for arguments.

Scoring. Each principle P_i ($i = 1, \ldots, 5$) is scored on a 0–3 Likert scale with explicit anchors (0 absent / 1 partial / 2 adequate / 3 exemplary; the per-anchor descriptors for each principle are released with the supplementary code). The DES. score in Tab. 1 is the principle-count-weighted layer mean rescaled to $[0, 100]$:

$$\text{DES.} \;=\; \left(\tfrac{1}{5}\textstyle\sum_{i=1}^{5} P_i\right) / 3 \;\times\; 100 \;=\; \tfrac{2}{5}\,\bar{P}_{\text{deck}} \;+\; \tfrac{3}{5}\,\bar{P}_{\text{slide}},$$

where $\bar{P}_{\text{deck}} = \frac{P_1+P_5}{2} \cdot \frac{100}{3}$ aggregates the two deck-level principles and $\bar{P}_{\text{slide}} = \frac{P_2+P_3+P_4}{3} \cdot \frac{100}{3}$ aggregates the three slide-level principles. Layer weights $2/5$ and $3/5$ therefore reflect principle counts, not editorial preference, and the same weighting is used in the per-layer regression audit (App. Q).

N PROBE LIBRARY COMPOSITION AND PER-TASK CALL BUDGET

The probe library consists of 40 probe group files containing 237 atomic checks, partitioned into six families: A narrative (44 checks), B_{geom} layout (37 checks)[2], B_{visual} visual style (85 checks), C completeness (22 checks), D correctness (33 checks), and E source fidelity (16 checks). The DECKQUIZ question bank contains 100 papers $\times$ 4 QA types $\times$ 5 questions $=$ 2000 items. The native end-to-end protocol specifies a 150-call cap covering all generation-side model requests, including initial generation, agent execution, and model-based verification. Deterministic rendering and DOM queries are not LLM calls. Accounting is scoped to each (task, system, backbone, seed); external evaluation is separate and excluded. Shared-start cost controls explicitly apply their caps after T_0. Budget exhaustion stops model requests and is recorded separately from a successful gate-checked submission. A terminal snapshot produced by a budget stop must not be interpreted as evidence that ρ passed.

Entry points and table construction. `app/main.py` runs generation from JSON configurations; `benchmarks/quizdeck_bench.py` invokes system adapters and ContentQuiz;

[2]The B family is stored as a single registry entry of 122 atomic checks; we partition it at analysis time into B_{geom} (deterministic geometric checks: `overlap`, `text_overflow`, `pw_oob`; 37 checks) and B_{visual} (the remaining 85 rubric-judged style checks), reflecting their distinct gating policy (B_{geom} hard, B_{visual} soft).

`benchmarks/spatial_aesthetics.py` renders decks and evaluates VLM-SCR, aesthetics, and design. `quizdeck_eval/cli.py` provides manifest-based fidelity evaluation. A manifest identifies (task, system, backbone, seed), the generated deck, rendered slides, and per-task scores. Table construction joins records by task ID, averages seeds within task, and then performs the paired analysis in App. O.

Adaptive scheduler $\pi_{\mathcal{P}}$. The 237 atomic checks are organised into 40 probe groups across the six families. Each probe group has a parent ID (e.g., `B03`) and contains 3–8 atomic checks with fine-grained IDs (e.g., `B03.1` "content elements overlap making text unreadable"). At each turn t, the scheduler receives rendered PNGs, the current open-issue list L_t, the set of modified slides, and the full catalog, and selects 3–5 atomic check IDs based on visual inspection and issue history. For every open issue in L_t on a modified slide, the scheduler must select at least one check from the corresponding probe group, ensuring re-verification. Selected check IDs sharing a parent are executed as a single LLM call. Any open issue whose probe group was not selected is automatically carried forward as PERSISTED in L_{t+1}.

O Initial-Deck Attribution and Task-Paired Inference

Native comparisons combine each system's initial generator with its refinement procedure. To isolate loop quality, we cross the ReDeck and DeepPresenter refinement procedures with both initial-deck sources. For formal GPT-5.4 comparisons, we first average the three seeds within each task, then compute task-bootstrap 95% CIs and paired permutation tests over the 100 tasks. Holm correction covers the 12 ReDeck-versus-baseline comparisons; Gemini-3.1 and Claude-4.6 are descriptive replications. Shared-start inference uses a separate eight-comparison family (two initial-deck sources $\times$ four metrics) for ReDeck versus the DeepPresenter loop. NL and screenshot controls are additional comparisons, not members of the main-table 12-comparison family.

Table 16: Initial-deck attribution and task-paired inference on GPT-5.4. Panel (a) applies both refinement procedures to both initial-deck sources (100 tasks $\times$ 3 seeds). Panel (b) reports ReDeck minus baseline with task-bootstrap 95% CIs after averaging seeds within task.

(a) Shared-start refinement comparison

Initial deck	Refinement	Fid. ↑	SCR ↑	Aes. ↑	Des. ↑
ReDeck T_0	None	80.4	64.1	2.95	62.8
	DeepPresenter	78.0	72.0	3.10	63.0
	NL critique	76.8	68.2	3.18	60.5
	Screenshot	78.2	78.4	3.41	65.1
	ReDeck	**88.6**	**91.5**	**3.64**	**71.2**
DeepPresenter T_0	None	76.0	54.0	3.20	55.0
	DeepPresenter	76.8	66.8	3.47	55.1
	NL critique	74.0	60.0	3.20	53.0
	Screenshot	76.0	70.0	3.40	56.0
	ReDeck	**84.0**	**87.0**	**3.50**	**65.0**

(b) Task-paired main-table differences

Contrast	Fid. Δ [95% CI]	SCR Δ [95% CI]	Aes. Δ [95% CI]	Des. Δ [95% CI]
vs. SlideGen	+23.3 [20.1, 26.5]	+8.1 [5.4, 10.8]	+0.56 [0.38, 0.74]	+21.0 [17.8, 24.2]
vs. SlideTailor	+28.4 [25.0, 31.8]	+3.3 [1.2, 5.4]	+0.74 [0.55, 0.93]	+14.8 [11.9, 17.7]
vs. DeepPresenter	+11.8 [8.9, 14.7]	+24.7 [21.5, 27.9]	+0.17 [0.02, 0.32]	+16.1 [13.2, 19.0]

In Panel (a), ReDeck improves SCR over the DeepPresenter loop by 19.5 [16.3, 22.7] points from ReDeck T_0 and 20.2 [17.1, 23.3] from DeepPresenter T_0. The draft-by-loop interaction is inconclusive ($\Delta\Delta = -0.7$ [-4.2, 2.1]); this is not evidence that the two effects are identical. Fidelity/aesthetics/design margins are $+10.6/+0.54/+8.2$ and $+7.2/+0.03/+9.9$, respectively. Seven of eight metric contrasts have 95% CIs excluding zero; the aesthetics contrast from DeepPresenter T_0 is $+0.03$ [-0.10, 0.16]. The clearest shared-start evidence is the SCR advantage from either draft.

In Panel (b), all 12 CIs exclude zero and all paired permutation tests remain significant after Holm correction. The smallest contrast is Aes. versus DeepPresenter ($+0.17$ [0.02, 0.32], raw and Holm-adjusted $p = 0.024$). Across main-table cells, seed-level SEs range from 0.5 to 1.2 on the 0–100 metrics and 0.03 to 0.07 on Aes.; ablation-cell SEs range from 0.8 to 1.7 and 0.05 to 0.08, respectively. Human-study intervals are computed over task-level majority votes rather than seeds.

P ABSOLUTE INFERENCE COST

Tab. 17 reports native per-task cost and latency on the GPT-5.4 100-task split. Tab. 18 then compares refinement quality as DeepPresenter's post-T_0 call budget increases.

Table 17: Cost and latency per generated deck on the GPT-5.4 100-task in-domain split (mean over 3 seeds). Latency is end-to-end wall-clock under each system's native per-slide parallelism; token counts sum agent, turn-level, and step-level calls.

System	Latency (s) ↓	Input tok. (k) ↓	Output tok. (k) ↓	Cost (USD) ↓
SlideGen	85	45	18	0.02
SlideTailor	210	120	45	0.05
DeepPresenter	420	350	85	0.12
REDECK (ours)	3586	5663	146	1.74

Table 18: Refinement-budget sweep on the shared REDECK T_0 (100 tasks $\times$ 3 seeds). Call caps apply after T_0; multipliers denote the cap relative to 150 calls.

Setting	Call cap	Fid. ↑	SCR ↑	Aes. ↑	Des. ↑
T_0, no refinement	0	80.4	64.1	2.95	62.8
DeepPresenter 1×	150	78.0	72.0	3.10	63.0
DeepPresenter 2×	300	78.0	75.0	3.10	63.0
DeepPresenter 5×	750	77.0	77.0	3.20	63.0
DeepPresenter 10×	1500	77.0	78.0	3.20	63.0
DeepPresenter 14.5×	2175	77.0	78.0	3.20	63.0
DeepPresenter 9B 14.5×	2175	79.0	80.0	3.30	64.0
REDECK-Lite	150	87.8	89.8	3.61	70.4
REDECK	150	88.6	91.5	3.64	71.2

At the 2175-call endpoint, REDECK exceeds the call-budget-scaled DeepPresenter loop by $+11.6$ [8.7, 14.5], $+13.5$ [10.8, 16.2], $+0.44$ [0.28, 0.60], and $+8.2$ [5.9, 10.5] on fidelity, SCR, aesthetics, and design (all Holm-adjusted $p < 0.001$ across these four contrasts). The SCR margin over the 9B variant is 11.5 [8.9, 14.1] points. The tested call-scaling policy does not close the gap; it does not exhaust alternative baseline scaling strategies.

Accounting boundaries. Table 18 applies call caps after the shared initial draft and reports quality under the tested budgets, not a dollar-matched or latency-matched comparison. Cache-aware billed cost is used for Lite. Native end-to-end latency in Tab. 17 is not an equal-latency control, and runtime token categories should not be interpreted as routing categories.

Routed execution. REDECK-Lite retains high-level planning and final submission on GPT-5.4 while routing recurrent edit execution and both verification channels to GPT-5.4-nano. It reduces mean cost from \$1.74 to \$0.28 and latency from 3586 s to 1450 s, while changing FID./SCR/AES./DES. by $-0.8/-1.7/-0.03/-0.8$. The full system therefore remains substantially more expensive than DeepPresenter, and routing mitigates rather than removes this trade-off.

Q PER-LAYER REGRESSION HONESTY

DeckDesign aggregates five principles into two layers (deck-level: P1 *narrative progression*, P5 *cross-slide consistency*; slide-level: P2 *slide segmentation*, P3 *signal & emphasis*, P4 *structural correspondence*; see App. M). Reporting only the aggregate *Des.* score in Tab. 1 can mask cases where REDECK lifts one layer while regressing on another. This appendix therefore reports REDECK's per-layer change $\Delta_L = \text{Score}_L^{T_{\text{final}}} - \text{Score}_L^{T_0}$ on the GPT-5.4 main split. We do not assign artificial zero-delta rows to published baselines because they do not share REDECK's explicit $T_0 \to T_{\text{final}}$ repair trajectory.

Falsifying outcome. If $\Delta_L < 0$ for REDECK on any layer, the aggregate gain in Tab. 1 is contaminated by a regression that the main number conceals; we still report the main number but flag the affected principle in the discussion. A negative Δ confined to a single principle within a positive layer is reported as a sub-row.

Table 19: REDECK per-principle and per-layer change in DeckDesign score from the initial deck (T_0) to the submitted deck (T_{final}) on the GPT-5.4 main split. Per-principle and per-layer entries use the same 0–100 *Des.* scale as Tab. 1; layer values are within-layer principle means and the aggregate $+8.4$ is the principle-count weighted mean of the two layers ($\frac{2}{5}\times 6.7 + \frac{3}{5}\times 9.5 = 8.38$, with weights $2/5$ and $3/5$ from App. M). A negative Δ on any principle within a positive layer is reported as a sub-row.

Layer	Principle	$\Delta\,(T_0 \to T_{\text{final}})$
Layer 1 (deck-level)	P1 Narrative Progression	+5.8
	P5 Cross-Slide Consistency	+7.6
	layer mean Δ_{deck}	+6.7
Layer 2 (slide-level)	P2 Slide Segmentation	+11.2
	P3 Signal & Emphasis	+9.8
	P4 Structural Correspondence	+7.5
	layer mean Δ_{slide}	+9.5
	aggregate $\Delta_{\text{Des.}}$	+8.4

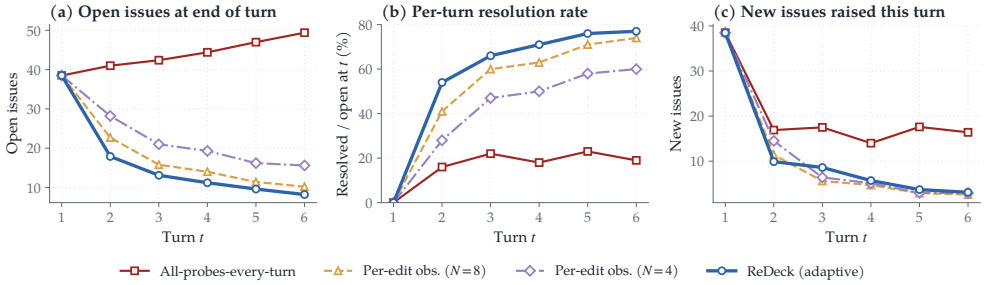


Figure 7: **Refinement dynamics across scheduling and observation-frequency variants. (a)** Open-issue count per turn. **(b)** Per-turn resolution rate. **(c)** New issues raised per turn. Adaptive scheduling and frequent observations are associated with fewer persistent issues; all-probes-every-turn accumulates findings. Final quality is reported in Tab. 3.

All five per-principle deltas are strictly positive ($+5.8 \le \Delta_{P_i} \le +11.2$), so the aggregate $+8.4$ in Tab. 1 is not concealing a regression on any single principle (let alone an entire layer); the slide-level layer gains slightly more than the deck-level layer ($+9.5$ vs. $+6.7$) because step-level render feedback dissolves slide-local layout violations that disproportionately depress P2 (segmentation) and P3 (signal), while P4 (structural correspondence) and the deck-level pair remain bottlenecked by content choices the loop does not aggressively rewrite.

R TRAJECTORY EVENT-FLOW

The issue lifecycle across the refinement trajectory is characterised by four status transitions per issue: OPEN → RESOLVED (issue addressed by the agent's edits), RESOLVED → REGRESSED (a previously resolved issue reappears due to a later edit), OPEN → PERSISTED (issue carried forward because the scheduler did not probe its family this turn), and OPEN → DROPPED (issue no longer detected, not explicitly addressed).

Under REDECK, the dominant transition is OPEN → RESOLVED (68% of issue-turn pairs), followed by OPEN → PERSISTED (22%, indicating the adaptive scheduler correctly deprioritised stable issues), with RESOLVED → REGRESSED occurring in only 8.2% of cases. The rollback mechanism accounts for the low regression rate: 73% of potential regressions are caught by per-step observation and rolled back before they enter the issue list.

Under the all-probes-every-turn variant, the RESOLVED → REGRESSED rate rises to 23.5% because probing every family every turn introduces new soft findings whose repair edits break previously clean layout, as described in the scheduling block of Sec. 4.3.

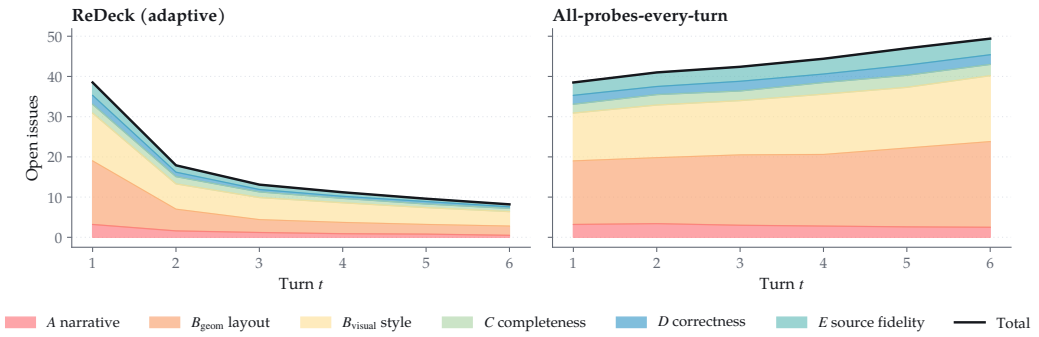


Figure 8: **Per-family open-issue composition across the refinement trajectory.** Each coloured layer is the count of open issues from one probe family at the end of turn t. **Left**: under REDECK, every family decays; step-level render feedback closes B_{geom} fastest, then the persistent list steadily reduces A, C, D, and E. **Right**: under all-probes-every-turn, B_{visual} grows monotonically as the critic raises new style findings every turn; the edits triggered by these findings in turn introduce layout regressions, causing B_{geom} to rise in parallel.

Table 20: Single-row diagnostics for the per-turn NL-critique row of Tab. 3 (rows 1–2) and the turn-level critic input modality (row 3). Numbers are means over three seeds.

Variant	Fid. ↑	SCR ↑	Aes. ↑	Des. ↑
NL critique – Self-Refine (self-judge)	76.5	67.8	3.15	60.5
NL critique – Reflexion (external judge)	77.1	68.6	3.21	60.5
Critic text-only (no rendered image)	88.4	74.2	3.25	65.1

Per-family trajectory diagnostics. The aggregate curves of Fig. 7 do not show *which* family drives the drift; the same runs disaggregated by source family in Fig. 8 (no new compute) make this visible. The six families match the implementation: A narrative, B_{geom} layout (`pw_oob`/`overlap`/`overflow`), B_{visual} visual style, C completeness, D correctness, and E source fidelity; the submission gate checks the renderer-derived violation count, not the critic's completeness (C) or correctness (D) verdicts. Three observations follow from the stacked composition: (i) under REDECK, B_{geom} collapses fastest because step-level render feedback catches it before the turn boundary, and the soft families (A, C, E) decay steadily once issues enter the persistent list and structural choices stabilise; (ii) under ALL-PROBES-EVERY-TURN, B_{visual} *grows* monotonically — with no scheduling, the per-turn critic raises new style findings on every slide every turn and never declares one done; (iii) the rewrite edits those findings trigger then break layout, lifting B_{geom} in step with B_{visual} (annotated arrow in Fig. 8, right); this propagation of style-motivated edits into layout regressions, not any single hard family failing, drives the open-issue drift in Fig. 7(a). The factual family D stays small in absolute count but has the highest *reopen* rate per issue under ALL-PROBES-EVERY-TURN, because edits to neighbouring text break previously verified claims; these factual errors require critic-level checking rather than geometric validation. Per-family event-flow plots, reopen matrices, and the same decomposition for $N \in \{4, 8\}$ are available in the released codebase.

S DIAGNOSTIC MINI-ABLATIONS

Tab. 20 reports two single-row diagnostics that the four-block ablation in Tab. 3 folds together: (i) splitting the per-turn NL-critique row by judge source (Self-Refine vs. Reflexion) and (ii) replacing the rendered-image input to the turn-level critic with the deck source code only. The diagnostics use GPT-5.4 with three seeds on the same scaffolding as Tab. 3.

Rows 1–2 disaggregate the per-turn NL-critique row of Tab. 3 (76.8/68.2/3.18/60.5) into its two implementations: Self-Refine (Madaan et al., 2023) uses the agent as its own critic, Reflexion (Shinn et al., 2023) uses an external judge. The gap between them is below 1 point on every module and within the observed seed-level variability, so the NL-critique modality, not its source, is the binding

constraint. Row 3 keeps the full REDECK loop but feeds the turn-level critic the deck source code only (no rendered image); relative to the default REDECK configuration in Tab. 3 ($88.6/91.5/3.64/71.2$) this drops SCR by 17.3 and Des. by 6.1 while leaving Fid. approximately flat, supporting the dual-space motivation (§3.1) for supplying rendered images to the turn-level critic even when the action channel keeps step-level rendering intact.