EXAM2TEACH: CAN LLM AGENTS TURN EXAM RESULTS INTO TEACHING MATERIALS?

Anonymous authors
Paper under double-blind review

ABSTRACT

High-quality teaching materials empower teachers to bridge learning gaps and provide highly targeted student support. Preparing such materials from exam results requires accurately interpreting student answers, translating identified needs into targeted instruction, and producing clear, coherent multimodal documents. In this work, we introduce `Exam2Teach`, a benchmark for evaluating whether autonomous LLM agents can generate high-quality, personalized teaching materials. `Exam2Teach` spans five core subjects shared by university-entrance curricula worldwide, using real exam papers, grading rubrics, de-identified student answers and scores. We design an open-source multi-agent framework for composing multi-step agent workflows. It powers our entire benchmark: agents generate documents, and evaluation workflows combine rule-based checks with agentic inspection of source evidence and rendered documents to assess *task completion*, *layout quality*, *factual correctness*, and *personalization*. The framework is general-purpose and extends readily to new tasks. An extensive evaluation of 10 agent configurations across 150 runs reveals uneven performance across four dimensions: the task completion threshold was met in 45.3% of runs, layout quality in 62.0%, factual correctness in 12.7%, and personalization in 26.7%. Only 7.3% passed all four dimensions, with no passing runs for review slides, and no single model consistently produced high-quality teaching materials of all categories. Even the strongest model overall, GPT-5.5, passed all four dimensions in only 40.0% of its runs. These results indicate a clear gap for frontier LLM agents in teaching material generation for AI-assisted education. Code and data are available at `https://anonymous.4open.science/r/Exam2TeachAnonymous-8E75/`.

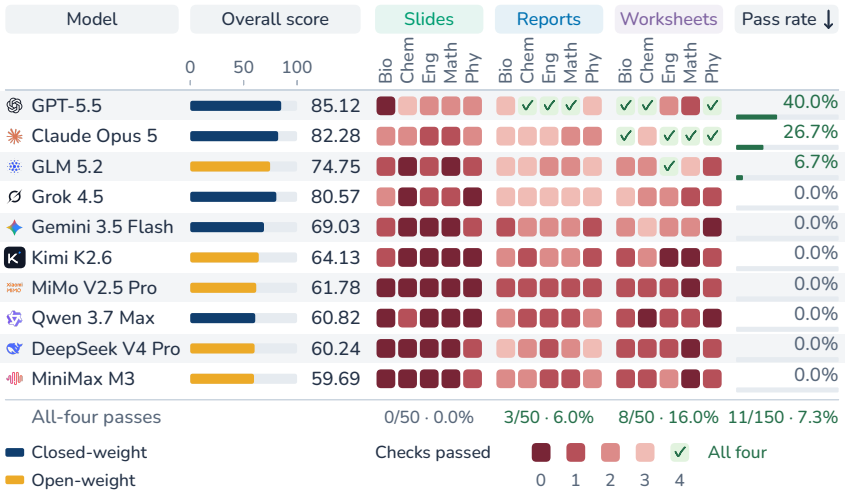


Figure 1: Overall scores and pass outcomes, ordered by descending pass rate, with ties broken by overall score. Each cell represents one model–task–subject run: darker red means fewer dimensions passed (0–3); green checks mark all-four passes. Rightmost bars show pass rates across 15 runs per model on a 0–100% scale.

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.

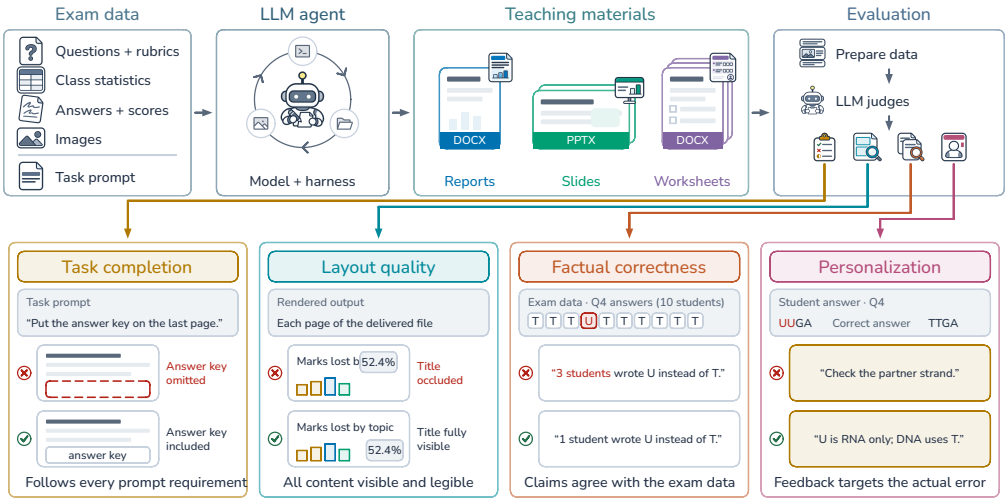


Figure 2: Overview of the Exam2Teach workflow. An LLM agent receives exam data and a task prompt, produces teaching materials, and is evaluated on task completion, layout quality, factual correctness, and personalization. The lower panels illustrate a failure and a corresponding correction for each dimension.

1 INTRODUCTION

Exam results reveal where a class struggles and what each student needs next (van Geel et al., 2016; Kannan et al., 2021), and turning this evidence into instruction is central to formative assessment (Black & Wiliam, 1998; Sadler, 1989; Hattie & Timperley, 2007). Through iterative feedback from over 50 frontline teachers across 15 schools, we identified three types of materials teachers need after every examination: (i) **class reports** that interpret performance and prioritize teaching, (ii) **review slides** that explain common mistakes, and (iii) **personalized worksheets** that give each student individual feedback and tailored exercises (Section 2). Teachers find these materials time-consuming to produce, and preparing a personalized worksheet for every student is often infeasible (e.g., in all 15 schools we work with). Autonomous LLM agents such as OpenAI Codex (OpenAI, 2026a), which plan, use tools, and revise their actions from results (Yao et al., 2023; Yang et al., 2024; Wang et al., 2025a), appear to be well suited to this work. However, there is a lack of systematic evaluation of LLM agents' ability to produce useful teaching materials across document types (see related work in Section 6). Therefore, this paper attempts to answer the following question: *Can autonomous LLM agents generate high-quality, personalized teaching materials from exam results?*

We introduce Exam2Teach (Figure 2), which evaluates three tasks corresponding to the three materials teachers identified, each delivered as an editable file: a DOCX class report, a PPTX review deck, and one DOCX worksheet per student. It covers five core subjects shared by university-entrance curricula worldwide, using real exam data (Section 2). Generated materials are evaluated on four dimensions: **task completion**, coverage of required content; **layout quality**, readability of rendered pages; **factual correctness**, whether textual and visual claims match the source materials; and **personalization**, alignment of explanations and next steps with the class's or student's learning needs. The evaluation combines rule-based checks with agentic inspection of source evidence and rendered documents. For each dimension, we set a pass threshold based on teachers' requirements, corresponding to the level at which a document needs only minor corrections before classroom use. Human validation shows that the evaluators closely agree with human judgments (Appendices F.2, G.2, H.2, and I.2). We also design a multi-agent framework for composing multi-step agent workflows, which powers both document generation and evaluation. It runs generation and evaluation pipelines from a single configuration and extends to new models, exams, and tasks.

We evaluate 10 agent configurations on the three tasks and five subjects. Each run pairs one configuration with one task and one subject, giving 150 runs and 700 output documents, and a run passes only if it meets the thresholds on all four dimensions. As shown in Figure 1, only 11 runs (7.3%) pass, and even the best configuration, GPT-5.5 with Codex, passes only 6 of its 15 runs (40.0%). High task completion does not guarantee usable materials: Claude Opus 5 meets the task completion threshold in all 15 runs, but only 4 of them pass (Section 5.1). Factual correctness is the weakest

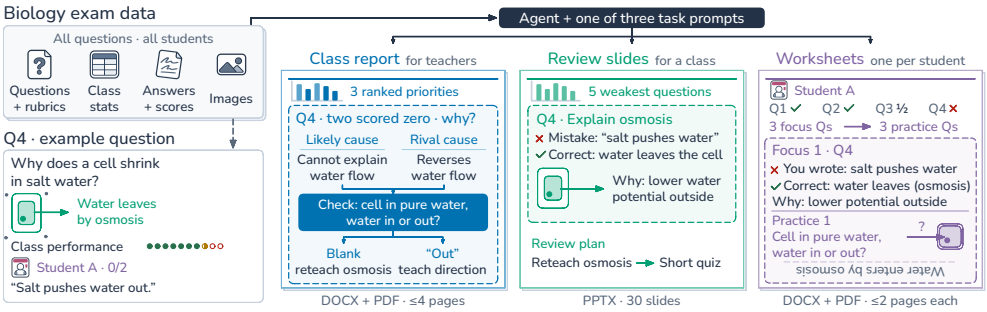


Figure 3: From one Biology exam dataset to three teaching documents. The agent receives all questions and student answers with one of three task prompts. Q4 is used for illustration.

dimension: only 12.7% of runs pass factual correctness, and incorrect quantitative claims account for 41.3% of classified factual errors (Tables 1 and 14). Review slides are the most difficult task, as agents struggle to place content on a fixed-size slide canvas: no slide run passes, and only 4 of 50 meet the layout quality threshold (Table 1). Output documents are often detached from students' actual answers: 92.9% of them contain at least one personalization misalignment (Table 29). Finally, feedback-based revision helps but is not sufficient: it improves the scores of all three tested agents on review slides, but only one revised deck passes all four dimensions (Section 5.3, Figure 10).

2 Tasks and Data

Exam2Teach turns the three teacher needs identified in the introduction into three generation tasks over one shared exam dataset: **class reports**, **review slides**, and **personalized worksheets** (Figure 3). The three tasks share the same exam data, and are each specified by one prompt stating the audience and required document. Each task is a long chain of steps and computations: an agent reads student answers, maps each answer to its question/rubric and student, computes statistics, diagnoses errors, and produces and checks an editable document that teachers can revise and use. Full prompts and detailed requirements are given in Appendices J and D.

Class reports: supporting teaching decisions. Schools require post-exam reports to guide teaching and inform internal discussions. Teachers need to understand students' learning needs and judge what to do next. Because report formats differ across schools, our task requires the shared analytical work in a concise report that teachers can verify quickly and transfer to their own formats. The agent produces an editable DOCX report within four A4 pages, a limit that forces prioritization of content selection. The report summarizes class performance and ranks up to three teaching priorities for teachers' follow-up.

Review slides: explaining common mistakes in class. The agent produces an editable 30-slide PPTX for classroom review. It summarizes class performance, selects and explains five important questions with high average error rates, diagnoses recurring problems across them, and ends with a short review plan (what to re-teach first and how to check whether it works). Representative student answers must be turned into corrected answers aligned with the grading rubric, explaining both the mistakes and the reasoning needed to avoid them. Statistics, formulas and diagrams, where the subject needs them, also have to fit on fixed-size slides and be clearly presented.

Personalized worksheets: addressing individual learning gaps. This task tests per-learner support. The agent produces one editable DOCX worksheet per student, at most two pages long, opening with the student's details and score overview. It offers three focus questions; for each, it provides the student's answer, the correct answer, an explanation of the student's error and how to correct it, the skill tested, and one matched practice question with an answer key. Focus questions are chosen by a fixed rule adapted from student–problem caution analysis (Harnisch & Linn, 1980) that favors student-specific gaps. Details are given in Appendix J.

Exam data and subject coverage. We use real exam papers, their grading rubrics, and a sample of de-identified candidate answers for each exam, from a university-entrance examination system. The data also report a classification of the candidates' scripts into five levels, but not actual marks, and thus we employed *human experts to mark every answer* using the rubrics. The same sources later serve as the **ground truth** for checking the generated materials (Section 4).

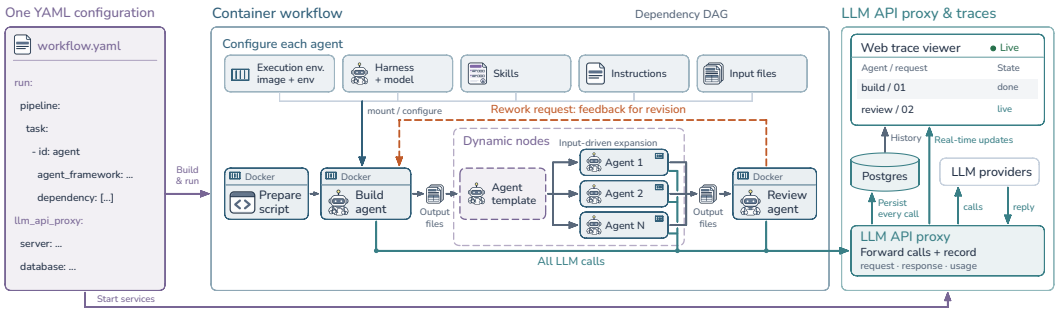


Figure 4: Our multi-agent framework.

The five subjects, Mathematics, Physics, Chemistry, Biology, and English Language, differ in the form of student work, language, and assessment structure. Each task's prompt is used unchanged for all five subjects, without subject-specific instructions (details in Appendix D). For each subject, we use one assessment and a constructed *class* of 10 candidates (20 for English), 60 records in all. Five assessments across three tasks yield 15 task instances, with each worksheet instance requiring a worksheet for every student; data composition and preparation are in Appendix D.

3 A Multi-Agent Framework

A core contribution of our work is a multi-agent framework for building agent pipelines (Figure 4). The framework is designed to make multi-agent workflows easy to design, test, and evaluate. It supports several commonly used agent harnesses, and a single YAML configuration is enough to define a workflow. The configuration declares each agent (its harness, execution environment, model, instruction, and skills) and the dependency graph among agents. The framework then builds the pipelines and runs them automatically. Changing a harness, model, or skill set therefore requires only a change to the configuration, so different workflow designs can be compared directly in the framework. The framework supports complex workflows, including dynamic nodes, where a single node expands at runtime into multiple agents according to its input parameters, and rework requests, where a downstream agent gives feedback to an upstream agent and asks it to revise its output. It also includes an LLM API proxy that records every model call, and a web interface that shows these records in real time, so agent traces can be inspected and analyzed while a workflow runs. This makes it easy to follow how each agent executes, locate failures, and optimize the workflow.

With this flexible framework, users can implement both the generation pipeline and the evaluation pipeline with little effort, as in Exam2Teach. We will open source the entire benchmark including this framework so that researchers and practitioners may use it to implement customized agent workflows easily and evaluate their performance, and obtain insights to improve them through an iterative evaluate-and-refine process within the same framework.

4 The Evaluation Method

We evaluate each output document on four dimensions identified from failures observed in early trials (Appendix E.1): **task completion**, **layout quality**, **factual correctness**, and **personalization**. Figure 5 depicts our evaluation pipeline and we give the details in the following subsections.

4.1 Data Preparation

We first prepare the inputs for the evaluation pipeline (Figure 5, left). The **exam data** are the input files the agents receive (Section 2). Agents deliver **editable documents** (also called **output docs**): DOCX reports/worksheets and PPTX slide decks, each also with a PDF export. These **PDF renderings** serve as the document file for task completion and are rasterized into **page images**, for visual judgment. GPT-5.5 transcribes the text in the page images into ordered **text blocks**. Charts, tables, or diagrams are not transcribed but marked in place by a **visual-object marker** pointing to their page images. A second pass replaces each marker with a description, giving **expanded**

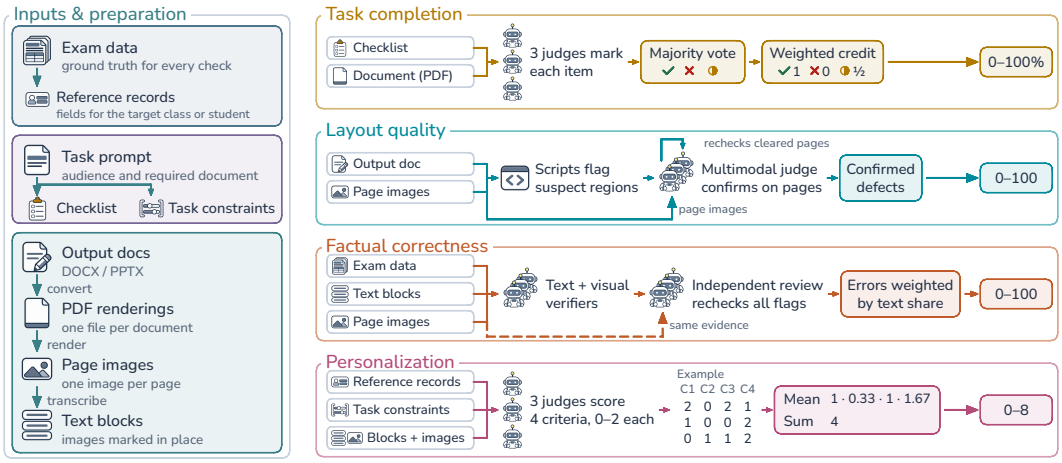


Figure 5: The evaluation pipeline. The left column shows how the inputs are prepared (Section 4.1). Stacked judges denote several parallel LLM agents.

text blocks for pure text-based evaluation. For personalization, we fix **task constraints**, a short restatement of what the task prompt requires and whether the task is for a specific class or student. We also use a script to select **reference records**, which are the fields of the exam data that concern that target class or student. Appendix E.2 gives rendering, block construction, and record details.

4.2 Pass Rates on Evaluation Criteria

Each run yields one score per dimension (Figure 5), and the run **passes** that dimension only if the score meets its threshold; a run **passes** only if it passes all four.

Frontline teachers emphasized materials with essentially no missing parts, minimal visual defects that can be quickly fixed, no major factual errors, and teaching options and worksheets tightly aligned with the specific needs of the target class or individual students.

To reflect this standard, we set strict **pass thresholds**: 95% for task completion, $90/100$ for layout quality, $90/100$ for factual correctness, and a personalization mean of at least 1.67 out of 2 averaged over the four personalization criteria, with each criterion mean at least 1.33 (details in Appendix I.1.1). These thresholds are set based on iterative teacher feedback rather than empirically calibrated boundaries. We mandate 95% task completion because fixing omitted requirements is highly demanding, often forcing teachers to create new content while verifying its layout quality, factual correctness, and personalization. We set the other thresholds to ensure that reasonably quick corrections are acceptable without disqualifying otherwise usable materials.

We also provide an **overall score**, which is a composite measure derived from the four evaluation dimension scores to capture an *agent's comprehensive capability* (details are given in Appendix E.3). The overall score allows for trade-offs among dimensions, while passing all four treats each dimension as a *non-negotiable minimum standard*. To clarify the specific technical demands each criterion imposes on agents, we unpack their underlying requirements below.

Task completion measures how fully an output doc meets the content requirements and structural constraints in the task prompt. For each task, we turn its prompt into one weighted checklist, one item per requirement. Each judge receives the checklist (without the weights) and the output doc. Three LLM judges (Claude Opus 5, GPT-5.5, and Grok 4.5) independently render the pages rather than relying on extracted text alone, labeling each item as "complete", "incomplete", or "absent". An item takes the label chosen by a majority of the judges (a three-way split counts as "incomplete"); an "incomplete" or "absent" item is an **unmet requirement**. Items earn full, half, or no credit accordingly, weighted by the item's weight. The percentage of credit earned is the **task completion rate**. Appendix F.1 details the procedure and scoring.

Layout quality measures whether the pages of an output doc are visually well presented. It counts four kinds of visible defect: content clipped at a page edge, text overlapping other text or images,

content spilling out of its box or hidden behind another shape, and illegible text such as tiny or faint type, broken formulas, or dense blocks that are hard to follow. A finding must name the affected region and how it impedes reading. The evaluation pipeline receives the output doc for script diagnostics, and its page images (Section 4.1) for visual review. The scripts first flag candidate defects from the output doc's geometry. A multimodal judge (Gemini 3.7 Flash) then inspects every page image, confirming or rejecting each flag and searching for defects the scripts missed. Confirmed, deduplicated findings are **layout defects**, converted by a fixed rule into a 0–100 layout quality score, with extra penalties for defects that recur across pages. Appendix G.1 gives the procedure and scoring rule.

Factual correctness measures whether the claims an output doc makes are supported by the task's exam data, directly or through valid calculation and reasoning. Claims beyond those sources are checked against the output doc's stated premises and reliable common knowledge. GPT-5.5 verifiers check the textual and visual claims against the exam data and the prepared text blocks and page images (Section 4.1). An independent review pass by GPT-5.5 rechecks every flagged claim against the same inputs and searches for missed errors. Each confirmed **factual error** carries a correction verified against the input data. We sum up (A) the sizes of all the blocks with confirmed errors, (B) the sizes of the blocks with only unverifiable claims, and (C) the sizes of all blocks, and then factual correctness is scored as $100 \times (1 - \frac{A+0.5B}{C})$. A single confirmed error in a large block can significantly affect the score, as this penalty proportionally reflects the amount of effort required for a teacher to check the whole affected block. Appendix H.1 gives the pipeline and formula.

Personalization evaluates whether an output doc's explanations, priorities, and next action items are tailored to students' specific learning needs. For reports and slides, this requires helping educators understand and instruct a *target group* based on collective learning needs, such as common misconceptions. For worksheets, it entails customizing explanations, practice questions, and guidance to *target individual* requirements. Shared content is appropriate when learners share needs. Three LLM judges (Claude Opus 5, GPT-5.5, and Grok 4.5) receive the output doc's page images and text blocks, the task constraints, and the target's reference records (Section 4.1). Each judge scores four criteria from 0 (generic or mismatched) to 2 (evidence-based tailoring with at most minor weaknesses): **target alignment**, whether the output doc fits the target's specific needs and level as reflected in their exam performance; **explanatory grounding**, whether key mistakes in the target's exam answers receive correct and sufficient explanations; **focus**, whether attention goes to the target's learning priorities (e.g., major areas where marks were lost); and **tailored actionability**, whether the output doc offers teachers effective teaching plans suited to the target class, or gives each target student actionable, customized learning advice. Judges also list evidence-linked issues (audited in Appendix E.4); those that should be fixed before use are **personalization misalignments** (Appendix I.1.1). Personalization scores are first averaged across judges within each criterion and then summed to a 0–8 score. Appendix I.1 gives the rubric and reference records and our repository provides the prompts.

5 Evaluation and Analysis

Setup. We evaluate 10 LLM-based agents using five closed-weight models: Claude Opus 5, GPT-5.5, Gemini 3.5 Flash, Grok 4.5, and Qwen 3.7 Max; and five open-weight models: DeepSeek V4 Pro, GLM 5.2, Kimi K2.6, MiMo V2.5 Pro, and MiniMax M3. Claude Opus 5 runs in Claude Code and GPT-5.5 in Codex, the first-party agent harnesses from their respective developers. The other eight models run in OpenCode, an open-source, model-agnostic harness, as a common setting (Anthropic, 2026; OpenAI, 2026a; OpenCode, 2026). Each reported result therefore describes a model–harness configuration. Further details on the generation setup, inputs, skills, prompts, and attempt limit appear in Appendix J.

5.1 Main Results

Table 1 presents the main-experiment results. Overall scores and per-task results for each model are reported in Appendix A.

High task completion does not guarantee deployment-ready teaching materials. Nearly half of the 150 runs, 45.3%, pass task completion, but only 7.3% pass all four dimensions. The gap

Table 1: Main-experiment pass rates and average generation cost. The upper block ranks all ten models by the proportion of their 15 runs passing all four criteria; ties are broken by the overall score (Eq. 1). The lower block aggregates results across all ten models for each task and for all tasks combined. Each pass entry shows passed/evaluated runs; values in parentheses are percentages. The Avg. cost ($) column reports USD per run.

Model / task	Task compl.	Layout	Factual corr.	Personal.	All pass	Avg. cost ($)
Overall model ranking						
1. GPT-5.5	13/15 (86.7)	9/15 (60.0)	10/15 (66.7)	10/15 (66.7)	6/15 (40.0)	5.42
2. Claude Opus 5	15/15 (100.0)	10/15 (66.7)	4/15 (26.7)	11/15 (73.3)	4/15 (26.7)	21.01
3. GLM 5.2	12/15 (80.0)	9/15 (60.0)	1/15 (6.7)	6/15 (40.0)	1/15 (6.7)	2.91
4. Grok 4.5	8/15 (53.3)	12/15 (80.0)	1/15 (6.7)	7/15 (46.7)	0/15 (0.0)	1.43
5. Gemini 3.5 Flash	6/15 (40.0)	9/15 (60.0)	3/15 (20.0)	1/15 (6.7)	0/15 (0.0)	4.83
6. Kimi K2.6	4/15 (26.7)	9/15 (60.0)	0/15 (0.0)	0/15 (0.0)	0/15 (0.0)	1.83
7. MiMo V2.5 Pro	0/15 (0.0)	9/15 (60.0)	0/15 (0.0)	0/15 (0.0)	0/15 (0.0)	0.15
8. Qwen 3.7 Max	1/15 (6.7)	9/15 (60.0)	0/15 (0.0)	1/15 (6.7)	0/15 (0.0)	2.54
9. DeepSeek V4 Pro	4/15 (26.7)	9/15 (60.0)	0/15 (0.0)	3/15 (20.0)	0/15 (0.0)	0.64
10. MiniMax M3	5/15 (33.3)	8/15 (53.3)	0/15 (0.0)	1/15 (6.7)	0/15 (0.0)	0.78
Aggregate across all 10 models						
Slides	16/50 (32.0)	4/50 (8.0)	4/50 (8.0)	6/50 (12.0)	0/50 (0.0)	8.26
Reports	33/50 (66.0)	50/50 (100.0)	3/50 (6.0)	20/50 (40.0)	3/50 (6.0)	1.93
Worksheets	19/50 (38.0)	39/50 (78.0)	12/50 (24.0)	14/50 (28.0)	8/50 (16.0)	2.27
Overall	68/150 (45.3)	93/150 (62.0)	19/150 (12.7)	40/150 (26.7)	11/150 (7.3)	4.15

persists even for the strongest models: Claude Opus 5 passes task completion in all 15 of its runs but only four runs pass all four dimensions, and GPT-5.5 passes all four in only 6 out of its 15 runs. Factual correctness is the most critical weakness, as only 19 of the 150 runs pass this dimension. Grok 4.5 passes layout quality in 12 of its 15 runs, but factual correctness in only one. In summary, current agents can produce teaching materials that appear nearly complete or visually polished, but limited end-to-end success and their weakness in factual correctness could limit the materials' classroom reliability and usability. Besides, the high costs of some models also limit their practical value; e.g., estimated generation costs average US$21.01 per run for Claude Opus 5 and US$5.42 for GPT-5.5, not to mention that their outputs still require substantial human review and revisions before actual deployment.

Data processing remains a major capability bottleneck for all tasks. Data processing requires agents to accurately derive, summarize, and interpret quantitative information from noisy, heterogeneous source data. Quantitative errors account for 41.3% of classified findings (Table 14). Data processing also affects personalization. In worksheets, 55.5% of the personalization misalignments are mismatched diagnosis and feedback (Table 27). Such feedback either ignores what the learner actually answered or gets the diagnosis or advice wrong.

Agents struggle with spatial composition, especially for slides. For slide generation, spatial composition requires agents to choose the size and position of every object on a fixed canvas while accommodating text, charts, formulas, and diagrams. Only 4 of 50 slide runs pass layout quality, far fewer than the 50 of 50 report runs and 39 of 50 worksheet runs. Weak spatial composition can also cause factual errors: slides contain 58 chart and diagram representation errors in which incorrect visual encoding makes the displayed relationships diverge from the underlying data (Table 14).

Personalization exposes limits in multi-step reasoning. Personalization involves a sequence of linked steps: identifying the learner or group, diagnosing their needs from evidence, and turning that diagnosis into suitable guidance. Effective personalization requires agents to reason through each step. Among the three tasks, 92.9% of documents have at least one personalization misalignment (Table 29). The personalization scores (Table 22) separate the tasks further. Slides identify class needs better than they turn them into actionable teaching steps (1.79/2 for *target alignment* vs. 1.15/2 for *tailored actionability*), and worksheets select relevant focus areas better than they explain individual errors (1.82/2 for *focus* vs. 1.28/2 for *explanatory grounding*).

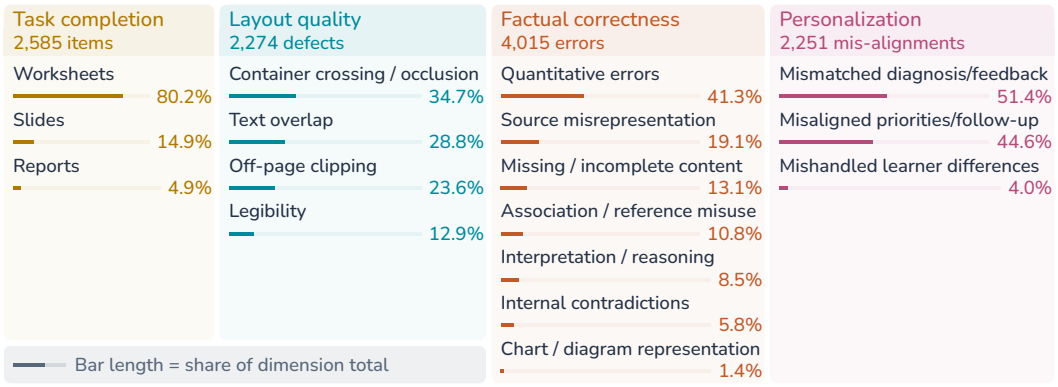
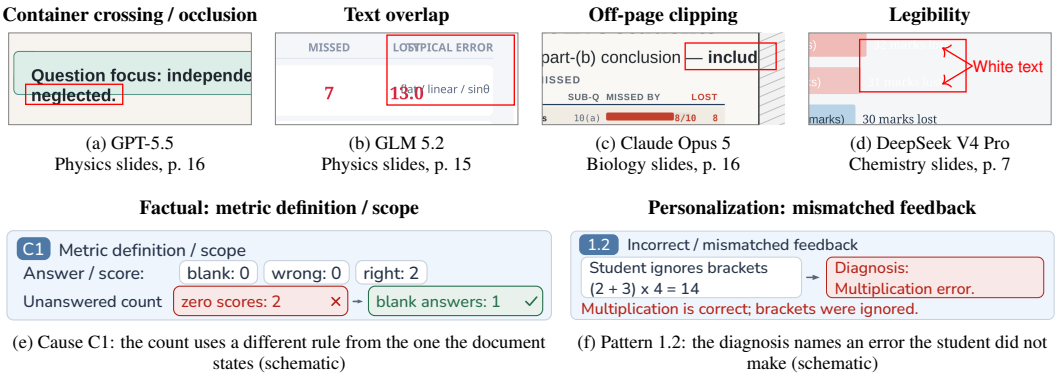


Figure 6: Failures across 50 slide decks, 50 reports, and 600 worksheets.



(a) GPT-5.5
Physics slides, p. 16

(b) GLM 5.2
Physics slides, p. 15

(c) Claude Opus 5
Biology slides, p. 16

(d) DeepSeek V4 Pro
Chemistry slides, p. 7

(e) Cause C1: the count uses a different rule from the one the document states (schematic)

(f) Pattern 1.2: the diagnosis names an error the student did not make (schematic)

Figure 7: Representative defects. (a)–(d): cropped slide pages showing the four most common layout-defect categories; red annotations mark the affected regions, and Appendix G.3 adds DOCX worksheet examples. (e)–(f): schematic illustrations of the most common quantitative-error cause and of a personalization pattern. Full illustrated catalogs are in Figures 35 and 49 in the appendix.

5.2 Failure Patterns

We report the % of failure patterns within each dimension, among 2,585 unmet requirements, 2,274 layout defects, 4,015 factual errors, and 2,251 personalization misalignments, as shown in Figure 6. Figure 7 shows representative failure examples.

Data processing bottlenecks. We classify the causes of quantitative factual errors. A specific cause can be identified for 893 findings (Figure 35). Metric or scope mismatches are the most common (53.0%): the calculation uses a different property, population, or inclusion rule from the one stated in the output doc, so correct arithmetic can still produce a false claim (Figure 7(e)). Incorrect numerical conditions or comparisons rank second (37.3%): a value supports the wrong conclusion when a threshold, ordering, or tie is applied incorrectly. Appendix H.4.6 details the classification of the four identifiable causes.

Spatial composition. Layout defects are tightly related to the format of an output doc: all 50 PPTX slide decks have at least one layout defect, compared with 3 of 50 DOCX class reports and 227 of 600 DOCX worksheets. PPTX fixes objects on a slide canvas, whereas DOCX can reflow content. Figure 7(a)–(d) shows cropped examples of each defect family in PPTX. Appendix G.3 adds DOCX cases. The examples suggest three related limitations: object placement can be imprecise on a fixed canvas; long text can wrap or overflow differently from what the agent anticipates; and formatting can ignore the rendered visual context or the document type. The latter includes low-contrast text and literal formula commands displayed in a worksheet.

Personalization misalignments. Figure 6 lists the three families of personalization misalignments. Diagnosis and feedback failures leave a student's actual mistake unexplained or replace it with

an unsupported explanation, so the learner cannot tell what to correct. Misdirected priorities and follow-up direct teaching or practice to a different need, even when the output doc initially identifies the right difficulty. Mishandled learner differences leave relevant groups without support or assign support to unsuitable learners. These problems can make an output doc appear personalized while substantially reducing its usefulness to the target class or student. Figure 7(f) shows one of the six underlying patterns, whereas Figure 49 defines and illustrates all six.

5.3 Ablation Studies

(1) Context length. We compare the default context limits with a smaller 128,000-token limit. The smaller limit has no uniform effect: the mean score for Claude Opus 5 with Claude Code rises from 77.77 to 81.90. For GPT-5.5, the score rises from 82.35 to 87.81 with Codex but falls from 82.13 to 56.45 with OpenCode, showing that the effect depends on harness context management. **(2) Agent harness.** Holding the model fixed, GPT-5.5 scores 87.65 with Claude Code, 82.35 with Codex, and 82.13 with OpenCode. GLM 5.2 scores 77.45 and 77.01 with OpenCode and Claude Code, respectively, but 69.67 with Codex. These results highlight the importance of the model's adaptability to the harness. **(3) Feedback and revision.** Revision affects the evaluation dimensions differently. On Biology slides, it substantially improves layout quality for Claude Opus 5, GPT-5.5, and Grok 4.5, but Grok 4.5 loses 2.54 points in factual correctness. Appendix B provides setup details and results for the three ablation experiments.

6 Related Work

Agent benchmarks and evaluation. Agent benchmarks span reasoning and tool use (Liu et al., 2024; Mialon et al., 2024), web and app environments (Zhou et al., 2024; Trivedi et al., 2024), software and terminal (Jimenez et al., 2024; Merrill et al., 2026), and computer and workplace environments (Xie et al., 2024; Wang et al., 2024b; Xu et al., 2025), and measure task success and progress (Yao et al., 2025; Ma et al., 2024), human-relative time horizons (Kwa et al., 2025), and deliverable quality on professional tasks (Patwardhan et al., 2025). Finer-grained assessment uses rubrics and agent judges (Ye et al., 2024; Li et al., 2026; Zhuge et al., 2025), which raises questions of judge reliability and benchmark validity (Zheng et al., 2023; Lù et al., 2025; Zhu et al., 2025). In education and presentation, benchmarks and systems address tutoring (Macina et al., 2025; Maurya et al., 2025), educational visualization and slide generation (Ji et al., 2026; Ge et al., 2025; Zheng et al., 2025), teaching workflows and materials (Chen et al., 2026b; Yao et al., 2026), automated exam-report generation (Xie et al., 2025), and slide fidelity to source materials (Chen et al., 2026a). Joint evaluation across document types and quality dimensions remains limited for educational artifacts generated from exam data. Exam2Teach addresses this gap by evaluating agents that generate class reports, review slides, and personalized worksheets from exam results, scored on task completion, layout quality, factual correctness, and personalization.

Appendix K reviews other related work including **educational foundations and applications**, as well as **LLM agents and execution frameworks**.

7 Conclusions

We present Exam2Teach, a benchmark for evaluating whether autonomous LLM agents can generate high-quality, personalized teaching materials from real exam data, judged on task completion, layout quality, factual correctness, and personalization. Across 150 runs of 10 agent configurations, only 7.3% of runs pass all four dimensions, and even the strongest configuration passes only 40.0% of its runs. Current agents can produce teaching materials that appear nearly complete or visually polished, but data processing and spatial composition remain major capability bottlenecks, with factual correctness as the most critical weakness, and personalization exposes limits in multi-step reasoning, as agents often produce plausible-sounding content detached from the target learner's or target group's actual needs. Our results indicate a clear gap for frontier LLM agents in generating teaching materials that are reliable and usable in classroom teaching. We also developed an open-source, multi-agent framework that builds and runs agent pipelines from a single configuration and records every model call. This framework allows new models, harnesses, and tasks to be evaluated with only configuration changes, and powers both generation and evaluation in Exam2Teach.

ETHICS STATEMENT

Source materials and release. The study uses public examination materials, including de-identified examples of candidate work provided by the examination authority for this research. The examination documents and the human-verified structured inputs derived from them are included in the anonymous repository (`https://anonymous.4open.science/r/Exam2TeachAnonymous-8E75/`) for the reproducibility of our results. However, original exam contents (e.g., handwritten answer scripts) are masked in the paper as we do not have permission to show such contents directly to the public.

Teacher consultations and benchmark data. Teachers informed the task specifications, quality expectations, and pass thresholds (Sections 2 and 4), and human experts marked the candidate answers that the benchmark uses as ground truth (Section 2). These contributors were consulted as domain experts, not studied as participants: no students, teachers, or schools were recruited as subjects, and no records about them enter the benchmark or its evaluation. Cohorts and learner labels are constructed from the examination authority's de-identified exemplars and their answers, scores, and performance levels. Failure examples use the same material and labels, and each excerpt was checked for names, candidate numbers, school names, and other direct identifiers before inclusion.

Intended use and risks. Exam2Teach evaluates systems that generate teaching and student feedback materials. Exam2Teach itself is not intended to be a grading system, nor to establish that a generated document is safe or effective for classroom use. Rather, Exam2Teach positions itself as a benchmark and provides an evaluation framework. Factual errors, misplaced learner evidence, and generic or inappropriate recommendations could misinform teaching or learners. Generated materials should be reviewed by a qualified human before use, and benchmark scores should not inform decisions about individual students.

REPRODUCIBILITY STATEMENT

Section 2 and Appendix D describe the source data, expert marking, cohort construction, and task requirements; Section 5 identifies the evaluated model–harness configurations; and Section 4 and Appendices E–I describe the evaluation procedure, scoring rules, and validation studies. Appendix J provides the generation setup, prompts, and agent skills, and Appendix B the ablation experiments. We release the code, configurations, prompts, checklists, output schemas, and the multi-agent framework for generation and evaluation, together with the examination documents and human-verified structured inputs, in an anonymous repository (`https://anonymous.4open.science/r/Exam2TeachAnonymous-8E75/`). Generation and LLM judging are stochastic and each instance is generated once per configuration (Section 5 and Appendix L), so reruns should expect variation in generated documents and scores. We remark that the raw examination data is provided strictly to ensure the reproducibility of our results, while we will release Exam2Teach including datasets for agent evaluation on our open-source repository instead.

AI USE STATEMENT

The research ideas, study design, and interpretation of the findings are the authors' own. We used generative AI tools to help implement the benchmark and evaluation framework from our designs, to critique drafts of the experimental design and evaluation protocol, and for literature search, figure preparation, drafting, and editing. AI assistance in preparing the source examination data was limited to optical character recognition. We did not use generative AI to generate synthetic datasets, propose or refine hypotheses, translate source materials, interpret results, or formulate mathematical claims.

LLMs are also part of the method. They extract and structure text from rendered documents, describe visual content, and evaluate task completion, layout quality, factual correctness, and personalization (Section 4; Appendix E), and they supported the qualitative analysis of evaluation findings by grouping failure patterns, inspecting suspected false positives, and preparing candidate claims and counterarguments for human review (Appendices H.4, I.2, and I.4).

The authors reviewed any AI-assisted writing and verified the citations. The authors reviewed and tested the AI-assisted code for correctness and ran the full generation and evaluation pipeline end to end. Human experts marked the source candidate answers using the examination rubrics (Section 2), and human checks of the automated evaluation, covering task-completion decisions, layout annotations, factual-error checks, and personalization claims, are reported with their coverage and limitations in Appendix E.4. These checks cover specified samples and do not establish that every automated judgment is correct. We take responsibility for the final content of this work, including all artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Claude Code overview. Product documentation, 2026. URL https://code.claude.com/docs/en/overview. Accessed September 18, 2026.

Paul Black and Dylan Wiliam. Inside the black box: Raising standards through classroom assessment. *Phi Delta Kappan*, 80(2):139–148, October 1998. URL https://eric.ed.gov/?id=EJ575146.

Paul Black and Dylan Wiliam. Developing the theory of formative assessment. *Educational Assessment, Evaluation and Accountability*, 21(1):5–31, February 2009. doi: 10.1007/s11092-008-9068-5. URL https://doi.org/10.1007/s11092-008-9068-5.

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, 2026a. URL https://arxiv.org/abs/2603.07244. Version 1.

Zixin Chen, Peng Liu, Rui Sheng, Haobo Li, Jianhong Tu, Xiaodong Deng, Kashun Shum, Dayiheng Liu, and Huamin Qu. TeachArena: Are language agents ready for realistic teaching work? arXiv preprint arXiv:2605.14322, 2026b. URL https://arxiv.org/abs/2605.14322. Version 3; under review.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL https://arxiv.org/abs/2309.11495.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Erkang Zhu, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, Peter Chang, Ricky Loynd, Robert West, Victor Dibia, Ahmed Awadallah, Ece Kamar, Rafah Hosn, and Saleema Amershi. Magentic-One: A generalist multi-agent system for solving complex tasks, 2024. URL https://arxiv.org/abs/2411.04468.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023. URL https://arxiv.org/abs/2210.08726.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and Trevor Darrell. AutoPresent: Designing structured visuals from scratch. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2902–2911, June 2025.

Jana Gonnermann-Müller, Jennifer Haase, Konstantin Fackeldey, and Sebastian Pokutta. FACET: Teacher-centred LLM-based multi-agent systems—towards personalized educational worksheets. arXiv preprint arXiv:2508.11401, 2025. URL https://arxiv.org/abs/2508.11401.

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. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.11738.

Delwyn L. Harnisch and Robert L. Linn. Analysis of item response patterns: Questionable test data and dissimilar curriculum practices. In *Annual Meeting of the American Educational Research Association*, Boston, MA, 1980. URL https://files.eric.ed.gov/fulltext/ED193265.pdf.

John Hattie and Helen Timperley. The power of feedback. *Review of Educational Research*, 77 (1):81–112, March 2007. doi: 10.3102/003465430298487. URL https://doi.org/10.3102/003465430298487.

Hugo He. PPT Master. GitHub repository, 2026. URL https://github.com/hugohe3/ppt-master. With contributors; accessed September 23, 2026.

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*, 2024. URL https://arxiv.org/abs/2308.00352.

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. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.01798.

Haonian Ji, Shi Qiu, Siyang Xin, Siwei Han, Zhaorun Chen, Dake Zhang, Hongyi Wang, and Huaxiu Yao. From EduVisBench to EduVisAgent: A benchmark and multi-agent framework for reasoning-driven pedagogical visualization. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=FVCpV04ZRe. arXiv:2505.16832.

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. arXiv:2310.06770.

Priya Kannan, Andrew D. Bryant, Shiyi Shao, and E. Caroline Wylie. Identifying teachers' needs for results from interim unit assessments. ETS Research Report RR-21-08, Educational Testing Service, 2021.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Chris Painter, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring AI ability to complete long software tasks. arXiv:2503.14499, 2025. URL https://arxiv.org/abs/2503.14499v4. Version 4, revised July 2026.

Sara H. Kyne, Martin M. H. Lee, and Charisse T. Reyes. Enhancing academic performance and student success through learning analytics-based personalised feedback emails in first-year chemistry. *Chemistry Education Research and Practice*, 24(3):971–983, 2023. doi: 10.1039/D3RP00032J. URL https://doi.org/10.1039/D3RP00032J.

Zhaohui Li, Peng He, Zhiyuan Chen, Honglu Liu, Zeyuan Wang, Tingting Li, and Jinjun Xiong. SciEval: A benchmark for automatic evaluation of K–12 science instructional materials. arXiv preprint arXiv:2604.25472, 2026. URL https://arxiv.org/abs/2604.25472.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. AgentBench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/e9df36b21ff4ee211a8b71ee8b7e9f57-Paper-Conference.pdf.

Xing Han Lù, Amirhossein Kazemnejad, Nicholas Meade, Arkil Patel, Dongchan Shin, Alejandra Zambrano, Karolina Stańczak, Peter Shaw, Christopher J. Pal, and Siva Reddy. AgentRewardBench: Evaluating automatic evaluations of web agent trajectories. In *Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2504.08942.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. AgentBoard: An analytical evaluation board of multi-turn LLM agents. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2401.13178.

Ziyu Ma, Hailang Huang, Shun Zou, Yong Wang, Shidong Yang, Yiming Hu, Fei Wei, and XiangXiang Chu. LongHorizon-Harness: Advancing long-horizon agents for real-world tasks, 2026. URL https://arxiv.org/abs/2608.01964.

Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. MathTutorBench: A benchmark for measuring open-ended pedagogical capabilities of LLM tutors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 204–221, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.11. URL https://aclanthology.org/2025.emnlp-main.11/.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.17651.

Kaushal Kumar Maurya, KV Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1234–1251, 2025. doi: 10.18653/v1/2025.naacl-long.57. URL https://aclanthology.org/2025.naacl-long.57/.

Kai Mei, Xi Zhu, Wujiang Xu, Mingyu Jin, Wenyue Hua, Zelong Li, Shuyuan Xu, Ruosong Ye, Yingqiang Ge, and Yongfeng Zhang. AIOS: LLM agent operating system. In *Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2403.16971.

Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Jenia Jitsev, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Jesse Hu, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-Bench: Benchmarking agents on hard, realistic tasks in command line interfaces. arXiv preprint arXiv:2601.11868, 2026. URL https://arxiv.org/abs/2601.11868v1. Version 1.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2311.12983.

OpenAI. Introducing the Codex app. Product release, February 2026a. URL https://openai.com/index/introducing-the-codex-app/. Published February 2, 2026; accessed September 18, 2026.

OpenAI. Build skills. Official Codex documentation, 2026b. URL https://learn.chatgpt.com/docs/build-skills. Accessed September 23, 2026.

OpenCode. OpenCode: Ai coding agent built for the terminal. Product documentation, 2026. URL https://opencode.ai/docs/. Accessed September 18, 2026.

OpenRouter. List all endpoints for a model. https://openrouter.ai/docs/api/api-reference/endpoints/list-all-endpoints-for-a-model, n.d. API documentation, accessed 2026-09-24.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems, 2023. URL https://arxiv.org/abs/2310.08560.

Abelardo Pardo, Jelena Jovanović, Shane Dawson, Dragan Gašević, and Negin Mirriahi. Using learning analytics to scale the provision of personalised feedback. *British Journal of Educational Technology*, 50(1):128–138, January 2019. doi: 10.1111/bjet.12592. URL https://doi.org/10.1111/bjet.12592.

Tejal Patwardhan, Rachel Dias, Elizabeth Proehl, Grace Kim, Michele Wang, Olivia Watkins, Simón Posada Fishman, Marwan Aljubeh, Phoebe Thacker, Laurance Fauconnet, Natalie S. Kim, Patrick Chao, Samuel Miserendino, Gildas Chabot, David Li, Michael Sharman, Alexandra Barr, Amelia Glaese, and Jerry Tworek. GDPval: Evaluating AI model performance on real-world economically valuable tasks, 2025. URL https://arxiv.org/abs/2510.04374.

D. Royce Sadler. Formative assessment and the design of instructional systems. *Instructional Science*, 18(2):119–144, June 1989. doi: 10.1007/BF00117714. URL https://doi.org/10.1007/BF00117714.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.11366.

The Document Foundation. Starting LibreOffice software with parameters. https://help.libreoffice.org/latest/en-GB/text/shared/guide/start_parameters.html, n.d. Online documentation, accessed 2026-09-24.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 16022–16076, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Marieke van Geel, Trynke Keuning, Adrie J. Visscher, and Jean-Paul Fox. Assessing the effects of a school-wide data-based decision-making intervention on student achievement growth in primary schools. *American Educational Research Journal*, 53(2):360–394, 2016. doi: 10.3102/0002831216637346.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In *Proceedings of the 41st International Conference on Machine Learning*, 2024a. URL https://arxiv.org/abs/2402.01030.

Xingyao Wang, Boxuan Li, Yufan Song, Frank F. Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang H. Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Yanjun Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, Junyang Lin, Robert Brennan, Hao Peng, Heng Ji, and Graham Neubig. Openhands: An open platform for AI software developers as generalist agents. In *International Conference on Learning Representations*, 2025a. URL https://arxiv.org/abs/2407.16741.

Zilong Wang, Yuedong Cui, Li Zhong, Zimin Zhang, Da Yin, Bill Yuchen Lin, and Jingbo Shang. OfficeBench: Benchmarking language agents across multiple applications for office automation, 2024b. URL https://arxiv.org/abs/2407.19056.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025b. URL https://arxiv.org/abs/2409.07429.

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*, 2024. URL https://arxiv.org/abs/2308.08155.

Donglin Xie, Xiang Li, Qingzhi Xiong, Sitong Wang, Yaling Jiang, Ye Liang, and Anan Jin. AutoEER: Large language model agent framework for automated english exam report. In *2025 7th International Academic Exchange Conference on Science and Technology Innovation (IAECST)*, pp. 6–10. IEEE, 2025. doi: 10.1109/IAECST68792.2025.11415007. URL https://doi.org/10.1109/IAECST68792.2025.11415007.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Jing Hua Toh, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5d413e48f84dc61244b6be550f1cd8f5-Paper-Datasets_and_Benchmarks_Track.pdf.

Frank F. Xu, Yufan Song, Boxuan Li, Yuxuan Tang, Kritanjali Jain, Mengxue Bao, Zora Z. Wang, Xuhui Zhou, Zhitong Guo, Murong Cao, Mingyang Yang, Hao Yang Lu, Amaad Martin, Zhe Su, Leander Melroy Maben, Raj Mehta, Wayne Chi, Lawrence Jang, Yiqing Xie, Shuyan Zhou, and Graham Neubig. TheAgentCompany: Benchmarking LLM agents on consequential real world tasks. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. arXiv:2412.14161.

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. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15793.

Huaiyuan Yao, Wanpeng Xu, Justin Turnau, Nadia Kellam, and Hua Wei. Instructional agents: Reducing teaching faculty workload through multi-agent instructional design. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics, Volume 1: Long Papers*, pp. 4087–4109. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.191. URL https://aclanthology.org/2026.eacl-long.191/.

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*, 2023. URL https://arxiv.org/abs/2210.03629.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2406.12045.

Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. FLASK: Fine-grained language model evaluation based on alignment skill sets. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=CYmF38ysDa.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. PPTAgent: Generating and evaluating presentations beyond text-to-slides. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 14402–14418, 2025. doi: 10.18653/v1/2025.emnlp-main.728. URL https://aclanthology.org/2025.emnlp-main.728/.

Jingsheng Zheng, Xinyuan Fang, Jintian Zhang, Zhengke Gui, Huajun Chen, and Ningyu Zhang. OneDayAgent: Towards a long-horizon harness for autonomous agents, 2026. URL https://arxiv.org/abs/2608.05013.

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, 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, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.13854.

Yuxuan Zhu, Tengjun Jin, Yada Pruksachatkun, Andy Zhang, Shu Liu, Sasha Cui, Sayash Kapoor, Shayne Longpre, Kevin Meng, Rebecca Weiss, Fazl Barez, Rahul Gupta, Jwala Dhamala, Jacob Merizian, Mario Giulianelli, Harry Coppock, Cozmin Ududec, Antony Kellermann, Jasjeet Sekhon, Jacob Steinhardt, Sarah Schwettmann, Arvind Narayanan, Matei A. Zaharia, Ion Stoica, Percy Liang, and Daniel Kang. Establishing best practices in building rigorous agentic benchmarks. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025. arXiv:2507.02825.

Yu Zhuang, Kefei Chen, Yitong Duan, Shuxin Zheng, Jian Li, and Xu-Yao Zhang. AgentRewind: Recoverable execution for long-horizon LLM agents, 2026. URL https://arxiv.org/abs/2608.14380.

Mingchen Zhuge, Changsheng Zhao, Dylan R. Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, Yangyang Shi, Vikas Chandra, and Jürgen Schmidhuber. Agent-as-a-Judge: Evaluate agents with agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80569–80611. PMLR, 2025. URL https://proceedings.mlr.press/v267/zhuge25a.html.

Appendix guide. Appendices A, B, and C present the full results, ablations, and generation usage and estimated cost. Appendices D and E describe the tasks and shared evaluation procedures. Appendices F–I provide methods, validation, detailed results, and failure examples for each dimension. Detailed generation setup and prompts appear in Appendix J; evaluator prompts and checklists are provided in our repository. Appendix K provides additional related work. Appendix L discusses limitations.

A Detailed Main-Experiment Results

Table 2 aggregates the five subjects for each task–model pair and reports the corresponding results across all three tasks. It shows scores and pass rates for the four evaluation dimensions, task and overall scores from Eq. 1, the rate of passing all four dimensions, and the average generation cost. Each run is one task on one subject, so a model row aggregates 5 runs in a task block and 15 in the overall block; the all-model rows aggregate 50 and 150, respectively. Pass thresholds are given in Section 4.2. Costs are estimated from producing-model token usage and per-token prices (Appendix C).

Table 2: Main-experiment results by task and model. Pass-rate cells show passed/evaluated task–subject runs followed by the percentage in parentheses. Personalization scores use a 0–8 scale; the other dimensions, task scores, and overall scores use 0–100 scales. The Score column gives the task score in each task block and the overall score in the overall block (Eq. 1). Cost is mean primary-model generation cost in USD per task–subject run.

Model	Task completion		Layout quality		Factual corr.		Personalization		Score	All four	Avg. cost
	Score	Pass rate	Score	Pass rate	Score	Pass rate	Score	Pass rate	Score	Pass rate	($)
Slides											
Claude Opus 5	99.46	5/5 (100%)	49.11	0/5 (0%)	73.63	0/5 (0%)	7.00	3/5 (60%)	70.06	0/5 (0%)	46.01
DeepSeek V4 Pro	88.38	1/5 (20%)	49.13	0/5 (0%)	61.17	0/5 (0%)	5.53	0/5 (0%)	52.99	0/5 (0%)	1.14
Gemini 3.5 Flash	91.35	2/5 (40%)	68.52	0/5 (0%)	66.19	0/5 (0%)	5.73	0/5 (0%)	62.61	0/5 (0%)	8.38
GLM 5.2	95.59	3/5 (60%)	65.12	0/5 (0%)	64.94	0/5 (0%)	6.00	0/5 (0%)	65.01	0/5 (0%)	4.94
GPT-5.5	94.96	3/5 (60%)	55.91	0/5 (0%)	91.49	4/5 (80%)	6.33	2/5 (40%)	73.23	0/5 (0%)	10.93
Grok 4.5	92.43	1/5 (20%)	87.79	2/5 (40%)	83.25	0/5 (0%)	6.53	1/5 (20%)	77.77	0/5 (0%)	2.60
Kimi K2.6	81.44	0/5 (0%)	72.33	1/5 (20%)	56.78	0/5 (0%)	5.53	0/5 (0%)	53.07	0/5 (0%)	3.19
MiMo V2.5 Pro	83.60	0/5 (0%)	82.13	0/5 (0%)	68.66	0/5 (0%)	5.00	0/5 (0%)	59.24	0/5 (0%)	0.25
MiniMax M3	88.83	1/5 (20%)	60.93	0/5 (0%)	57.09	0/5 (0%)	5.33	0/5 (0%)	54.29	0/5 (0%)	1.03
Qwen 3.7 Max	87.57	0/5 (0%)	82.23	1/5 (20%)	58.02	0/5 (0%)	5.40	0/5 (0%)	59.66	0/5 (0%)	4.09
All models	**90.36**	**16/50 (32%)**	**67.32**	**4/50 (8%)**	**68.12**	**4/50 (8%)**	**5.84**	**6/50 (12%)**	**62.79**	**0/50 (0%)**	**8.26**
Reports											
Claude Opus 5	99.06	5/5 (100%)	99.49	5/5 (100%)	65.52	0/5 (0%)	7.00	3/5 (60%)	81.53	0/5 (0%)	8.43
DeepSeek V4 Pro	96.04	3/5 (60%)	98.52	5/5 (100%)	52.53	0/5 (0%)	6.87	3/5 (60%)	73.29	0/5 (0%)	0.37
Gemini 3.5 Flash	95.10	3/5 (60%)	98.32	5/5 (100%)	57.41	0/5 (0%)	5.20	0/5 (0%)	68.43	0/5 (0%)	2.91
GLM 5.2	98.49	5/5 (100%)	100.00	5/5 (100%)	59.83	0/5 (0%)	6.87	3/5 (60%)	78.48	0/5 (0%)	1.51
GPT-5.5	98.87	5/5 (100%)	100.00	5/5 (100%)	90.98	3/5 (60%)	7.53	5/5 (100%)	93.57	3/5 (60%)	2.94
Grok 4.5	97.74	5/5 (100%)	100.00	5/5 (100%)	72.82	0/5 (0%)	7.20	5/5 (100%)	84.18	0/5 (0%)	0.69
Kimi K2.6	96.23	3/5 (60%)	100.00	5/5 (100%)	61.42	0/5 (0%)	5.93	0/5 (0%)	73.92	0/5 (0%)	0.63
MiMo V2.5 Pro	90.38	0/5 (0%)	100.00	5/5 (100%)	54.75	0/5 (0%)	5.40	0/5 (0%)	65.21	0/5 (0%)	0.08
MiniMax M3	95.66	3/5 (60%)	100.00	5/5 (100%)	30.04	0/5 (0%)	5.80	0/5 (0%)	61.00	0/5 (0%)	0.26
Qwen 3.7 Max	93.02	1/5 (20%)	100.00	5/5 (100%)	50.01	0/5 (0%)	6.27	1/5 (20%)	68.37	0/5 (0%)	1.48
All models	**96.06**	**33/50 (66%)**	**99.63**	**50/50 (100%)**	**59.53**	**3/50 (6%)**	**6.41**	**20/50 (40%)**	**74.80**	**3/50 (6%)**	**1.93**
Student worksheets											
Claude Opus 5	99.79	5/5 (100%)	100.00	5/5 (100%)	90.10	4/5 (80%)	7.84	5/5 (100%)	95.24	4/5 (80%)	8.58
DeepSeek V4 Pro	73.79	0/5 (0%)	94.98	4/5 (80%)	69.43	0/5 (0%)	4.67	0/5 (0%)	54.43	0/5 (0%)	0.40
Gemini 3.5 Flash	87.67	1/5 (20%)	95.03	4/5 (80%)	88.52	3/5 (60%)	6.08	1/5 (20%)	76.04	0/5 (0%)	3.19
GLM 5.2	96.36	4/5 (80%)	94.00	4/5 (80%)	76.60	1/5 (20%)	6.66	3/5 (60%)	80.76	1/5 (20%)	2.28
GPT-5.5	97.06	5/5 (100%)	96.20	4/5 (80%)	91.09	3/5 (60%)	6.92	3/5 (60%)	88.55	3/5 (60%)	2.39
Grok 4.5	95.13	2/5 (40%)	97.54	5/5 (100%)	78.87	1/5 (20%)	6.14	1/5 (20%)	79.75	0/5 (0%)	1.00
Kimi K2.6	87.58	1/5 (20%)	88.42	3/5 (60%)	68.68	0/5 (0%)	5.51	0/5 (0%)	65.39	0/5 (0%)	1.65
MiMo V2.5 Pro	80.36	0/5 (0%)	94.99	4/5 (80%)	74.10	0/5 (0%)	4.71	0/5 (0%)	60.90	0/5 (0%)	0.12
MiniMax M3	86.68	1/5 (20%)	87.19	3/5 (60%)	64.00	0/5 (0%)	5.82	1/5 (20%)	63.79	0/5 (0%)	1.06
Qwen 3.7 Max	78.71	0/5 (0%)	91.75	3/5 (60%)	55.47	0/5 (0%)	5.19	0/5 (0%)	54.44	0/5 (0%)	2.04
All models	**88.31**	**19/50 (38%)**	**94.01**	**39/50 (78%)**	**75.69**	**12/50 (24%)**	**5.95**	**14/50 (28%)**	**71.93**	**8/50 (16%)**	**2.27**
Overall											
Claude Opus 5	99.44	15/15 (100%)	82.87	10/15 (66.7%)	76.42	4/15 (26.7%)	7.28	11/15 (73.3%)	82.28	4/15 (26.7%)	21.01
DeepSeek V4 Pro	86.07	4/15 (26.7%)	80.88	9/15 (60%)	61.04	0/15 (0%)	5.69	3/15 (20%)	60.24	0/15 (0%)	0.64
Gemini 3.5 Flash	91.37	6/15 (40%)	87.29	9/15 (60%)	70.71	3/15 (20%)	5.67	1/15 (6.7%)	69.03	0/15 (0%)	4.83
GLM 5.2	96.81	12/15 (80%)	86.37	9/15 (60%)	67.12	1/15 (6.7%)	6.51	6/15 (40%)	74.75	1/15 (6.7%)	2.91
GPT-5.5	96.96	13/15 (86.7%)	84.04	9/15 (60%)	91.19	10/15 (66.7%)	6.93	10/15 (66.7%)	85.12	6/15 (40%)	5.42
Grok 4.5	95.10	8/15 (53.3%)	95.11	12/15 (80%)	78.31	1/15 (6.7%)	6.62	7/15 (46.7%)	80.57	0/15 (0%)	1.43
Kimi K2.6	88.42	4/15 (26.7%)	86.92	9/15 (60%)	62.29	0/15 (0%)	5.66	0/15 (0%)	64.13	0/15 (0%)	1.83
MiMo V2.5 Pro	84.78	0/15 (0%)	92.37	9/15 (60%)	65.84	0/15 (0%)	5.04	0/15 (0%)	61.78	0/15 (0%)	0.15
MiniMax M3	90.39	5/15 (33.3%)	82.71	8/15 (53.3%)	50.38	0/15 (0%)	5.65	1/15 (6.7%)	59.69	0/15 (0%)	0.78
Qwen 3.7 Max	86.43	1/15 (6.7%)	91.33	9/15 (60%)	54.50	0/15 (0%)	5.62	1/15 (6.7%)	60.82	0/15 (0%)	2.54
All models	**91.58**	**68/150 (45.3%)**	**86.99**	**93/150 (62%)**	**67.78**	**19/150 (12.7%)**	**6.07**	**40/150 (26.7%)**	**69.84**	**11/150 (7.3%)**	**4.15**

B Ablation Results

Each comparison uses one Biology input per task and agent configuration; the worksheet input covers ten students. Each contrast compares single outputs, so it does not capture run-to-run or cross-subject variation. Each ablation is run by the multi-agent framework described in Section 3, which keeps the inputs, skills, and prompts identical across the compared configurations.

Context length. We test Claude Code with Claude Opus 5, Codex with GPT-5.5, and OpenCode with GLM 5.2 on the Biology inputs for all three tasks: class reports, review slides, and student worksheets. We compare an explicit context limit of 128,000 tokens with the default context budgets used in the main experiment: 1,000,000 tokens for Claude Code with Claude Opus 5, 272,000 for Codex with GPT-5.5, and 1,000,000 for OpenCode with GLM 5.2. We also test OpenCode with GPT-5.5 under its default 272,000-token budget and the 128,000-token limit to examine how the harness affects performance under a shorter context window. Figure 8 reports the results; each cell shows the default-setting score followed by the score with the 128,000-token limit.

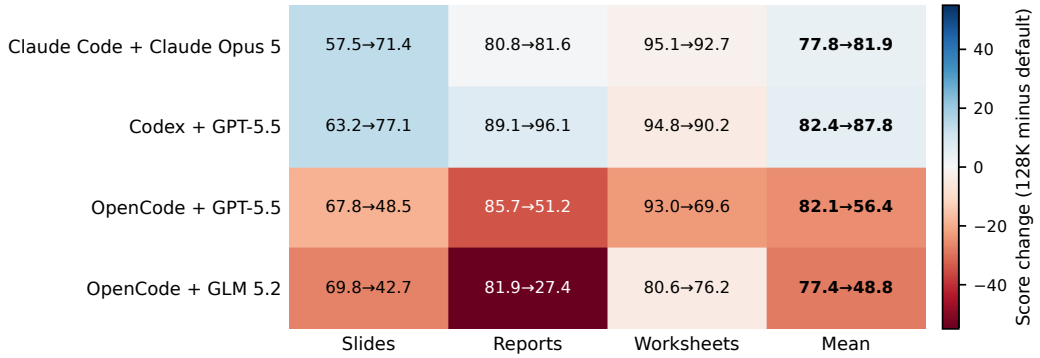


Figure 8: Biology task scores by agent configuration and context setting. Each cell shows default → 128,000-token context limit; color encodes the change (limited minus default). Mean weights the slides, reports, and worksheets equally.

Agent harness. We compare Claude Code, Codex, and OpenCode while holding the model fixed, using GPT-5.5 and GLM 5.2 on the Biology inputs for all three tasks. The context budgets follow the default lengths of the original Codex–GPT-5.5 and OpenCode–GLM 5.2 pairs: 272,000 tokens for GPT-5.5 and 1,000,000 tokens for GLM 5.2 across the three harnesses. Each harness runs with its own default configuration for other settings. Figure 9 reports the task scores and their equally weighted mean.

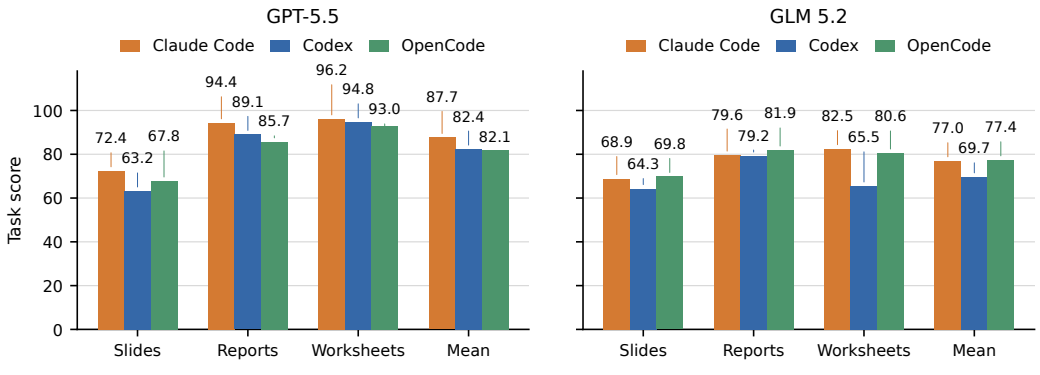


Figure 9: Biology task scores for GPT-5.5 and GLM 5.2 across three agent harnesses, with one output per task and configuration. Each bar uses the configuration of its named harness; Mean weights the slides, reports, and worksheets equally.

Feedback and revision. We test feedback-based revision only on Biology slides, using Claude Code with Claude Opus 5, Codex with GPT-5.5, and OpenCode with Grok 4.5. The agent first produces a slide deck, which is then checked on all four evaluation dimensions. Task completion and personalization use the same evaluation pipelines as the main experiment. Layout quality feedback comes directly from the script-based layout checker. For factual correctness, we divide the deck into batches of at most 10 pages and assign each batch to an independent Codex agent using GPT-5.5 to check against the source materials and provide feedback. This workload follows the factual-validity study in Appendix H.2.1, where Codex with GPT-5.5 achieves 93.3% mean recall on the 10-page condition, which we consider sufficient for generating revision feedback. After all four checks finish, any reported issues trigger a revision by the original producing agent. It receives its previous workspace, the original task instructions, and feedback organized by evaluation dimension, including issue descriptions and suggested corrections. The revised deck is checked again, and remaining issues produce another round of feedback. This loop allows at most two revision passes. The final deck is scored using the main-experiment evaluation pipelines. Figure 10 reports the results; each cell shows the original main-experiment score followed by the score after revision. The task score rises for all three agents. After revision, the GPT-5.5 deck passes all four dimensions; the Claude Opus 5 and Grok 4.5 decks pass task completion, layout quality, and personalization but still fail factual correctness. None of the three original decks passed all four.

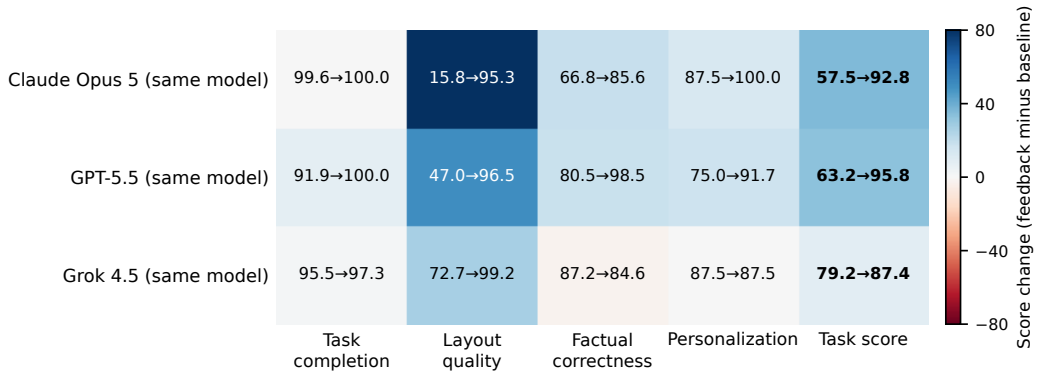


Figure 10: Biology slide scores before and after feedback-based revision. Each cell shows the original output → revised output for the same producing model and harness; color encodes the revised-minus-original change. For comparison, Personalization is rescaled from its original 0–8 scale to 0–100. The task score is calculated from the four evaluation dimensions.

C GENERATION TOKEN USAGE AND ESTIMATED COST

We record token usage for the 150 main-experiment runs (10 models, 3 tasks, 5 subjects). A run produces one report, one slide deck, or one batch of worksheets. We count only the producing model's tokens in the selected run; tokens used by other models, such as the vision helper, and by evaluation are excluded. Input tokens are uncached tokens plus cache reads, total tokens add output tokens, and no run records cache writes.

Costs are estimates, not billed amounts. For each model, we multiply its token counts by the official provider's price as listed on OpenRouter (OpenRouter, n.d.), taken from a 2026-09-24 09:56 UTC snapshot (Table 3). We use standard-tier rates without discounts, the highest rate when a price varies by time of day, base rates for GPT-5.5 and Grok 4.5, whose request-size surcharges cannot be applied to aggregated usage, and the DeepSeek V4 Pro-0813 endpoint as the price for DeepSeek V4 Pro. These are the costs in Table 2.

Table 3: OpenRouter pricing snapshot used for estimated generation costs, in US dollars per million tokens (2026-09-24 09:56 UTC). No selected run records cache-write tokens.

Producing model	Uncached input	Cache read	Output
Claude Opus 5	5	0.5	25
DeepSeek V4 Pro	1.32	0.044	3.96
Gemini 3.5 Flash	1.5	0.15	9
GLM 5.2	1.4	0.26	4.4
GPT-5.5	5	0.5	30
Grok 4.5	2	0.3	6
Kimi K2.6	0.95	0.16	4
MiMo V2.5 Pro	0.435	0.0036	0.87
MiniMax M3	0.3	0.06	1.2
Qwen 3.7 Max	1.475	0.295	4.425

Table 4: Generation-token usage by task, per subject–model run. All values are means except median total; M denotes million tokens.

Task	Input (M)		Output (M)	Total (M)	Median (M)	Cost ($)
	Uncached	Cache read				
Slides	0.74	16.65	0.15	17.53	11.55	8.26
Reports	0.22	3.99	0.04	4.24	3.04	1.93
Student worksheets	0.29	6.06	0.05	6.41	4.98	2.27

Slides use the most tokens and incur the highest estimated cost per run on average, followed by worksheet batches and reports (Table 4). The same order holds for median total tokens. Mean usage exceeds median usage for all three tasks, most markedly for slides. Across the three tasks, cache reads account for 94.8–95.8% of reported input tokens.

Table 5: Mean producing-model generation usage and estimated cost per run, by task and model. Totals use unrounded token counts.

Model	Input (M)		Output (M)	Total (M)	Cost ($)
	Uncached	Cache read			
Slides					
Claude Opus 5	0.75	64.99	0.39	66.13	46.01
DeepSeek V4 Pro	0.21	11.72	0.09	12.02	1.14
Gemini 3.5 Flash	2.16	19.59	0.24	22.00	8.38
GLM 5.2	0.41	13.92	0.17	14.50	4.94
GPT-5.5	0.45	12.09	0.09	12.63	10.93
Grok 4.5	0.41	4.33	0.08	4.82	2.60
Kimi K2.6	1.77	6.86	0.10	8.74	3.19
MiMo V2.5 Pro	0.35	10.42	0.07	10.84	0.25
MiniMax M3	0.14	14.23	0.11	14.48	1.03
Qwen 3.7 Max	0.71	8.34	0.13	9.19	4.09
Reports					
Claude Opus 5	0.20	10.02	0.10	10.31	8.43
DeepSeek V4 Pro	0.11	2.44	0.03	2.58	0.37
Gemini 3.5 Flash	0.63	7.40	0.10	8.12	2.91
GLM 5.2	0.16	4.39	0.03	4.58	1.51
GPT-5.5	0.18	2.47	0.03	2.68	2.94
Grok 4.5	0.21	0.61	0.01	0.84	0.69
Kimi K2.6	0.24	2.13	0.02	2.38	0.63
MiMo V2.5 Pro	0.12	3.40	0.02	3.54	0.08
MiniMax M3	0.07	3.59	0.02	3.67	0.26
Qwen 3.7 Max	0.25	3.43	0.02	3.71	1.48
Student worksheets					
Claude Opus 5	0.18	9.89	0.11	10.18	8.58
DeepSeek V4 Pro	0.12	3.27	0.02	3.42	0.40
Gemini 3.5 Flash	0.82	7.39	0.09	8.30	3.19
GLM 5.2	0.21	6.25	0.08	6.54	2.28
GPT-5.5	0.16	1.35	0.03	1.54	2.39
Grok 4.5	0.26	0.97	0.03	1.26	1.00
Kimi K2.6	0.55	5.93	0.05	6.52	1.65
MiMo V2.5 Pro	0.19	4.92	0.02	5.13	0.12
MiniMax M3	0.18	15.44	0.07	15.69	1.06
Qwen 3.7 Max	0.27	5.18	0.03	5.47	2.04

Across all three tasks, mean generation usage varies widely by model. At the high end, Claude Opus 5 averages 28.87 million tokens and US$21.01 per run. Qwen 3.7 Max provides a middle-range example at 6.12 million tokens and US$2.54, while Grok 4.5 uses the fewest tokens at 2.31 million and costs US$1.43.

C.1 Score–Cost Trade-offs

Figure 11 compares each model's overall score with its mean generation cost per run.

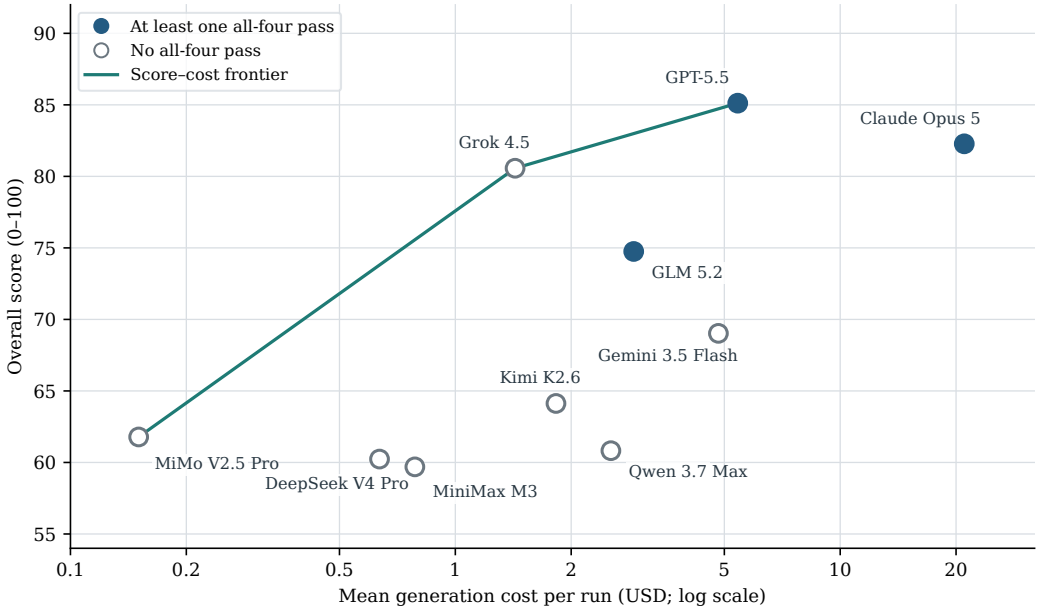


Figure 11: Main-experiment overall score versus mean generation cost per run. The cost axis is logarithmic; the line connects models on the score–cost frontier, and filled markers indicate at least one run passing all four dimensions.

MiMo V2.5 Pro, Grok 4.5, and GPT-5.5 form the score–cost frontier. MiMo V2.5 Pro has the lowest mean cost at US$0.15, but scores 61.78 and has no fully passing run. Grok 4.5 reaches 80.57 at US$1.43, also without a fully passing run. GPT-5.5 scores highest at 85.12 with a mean cost of US$5.42; it is the only frontier model with fully passing runs. Claude Opus 5 also has fully passing runs, but its score of 82.28 comes at a higher mean cost of US$21.01. The figure thus shows that lower cost, higher average quality, and fully passing outputs do not coincide in a single model.

D Exam Data and Task Details

D.1 Data composition and subject coverage

Section 2 describes the source materials, expert marking, and cohort sizes. The 60 records cover 120 exam questions, comprising 304 separately scored items, and 3,450 scored answers. The 304 scored items count only the smallest scored units: a question with subparts is counted through those subparts, not again as a whole.

Each class is constructed so that its candidates' records span the examination's five reported attainment levels, from lowest to highest. Each candidate record holds one candidate's complete set of answers. The candidates need not have attended the same class, but together they support analysis of shared difficulties and individual differences.

The supplied statistics include score rates, score distributions, and counts of full-, partial-, and zero-mark answers; their exact contents vary across assessments. Records also retain source-page references and associated visuals.

Subject and language coverage. Four subjects use English questions and candidate answers; Biology uses source material in a language other than English. All task outputs are in English, so Biology requires English explanations grounded in non-English evidence. Table 6 describes the evidence and production demands of each subject.

Table 6: The five subject assessments and their evidence and production demands.

Subject	Evidence and production demand
Mathematics	Dense notation, multi-step derivations, tables, and geometry whose symbols and logical dependencies must survive extraction, reasoning, editable authoring, and rendering.
Physics	Equations tied to units, sign conventions, and reported precision; diagrams, signed graph interpretation, and graph or sketch tasks whose answers depend on their working and physical assumptions.
Chemistry	Subscripts, superscripts, ionic charges, state symbols, balanced equations, electron diagrams, structural formulas, experimental set-ups, and graph interpretation or sketching.
Biology	Non-English questions and answers explained in English; micrographs, plots, tables, and diagrams to interpret, complete, plot, or draw and label without losing scientific meaning or visual references.
English Language	Mostly constrained reading items, including multiple choice, true/false/not-given, short answers, flow or table completion, accepted alternatives, and B1/B2 route mapping.

English Language assessment sections. The English Language reading assessment requires all candidates to complete a common section and choose one of two alternative sections of different difficulty, B1 (easier) and B2 (harder). The provider equates marks across the alternative sections using performance on the common section; the easier route has an attainment ceiling, while the harder route permits the full range of attainment levels. Agents must identify each student's section choice, sometimes from answer structure rather than an explicit learner field. The English Language class holds 20 candidates so that both sections are represented. Our benchmark uses expert-assigned item marks for instructional analysis. For worksheet selection, the comparison group comprises classmates who chose the same alternative section (B1 or B2), including when the selected question comes from the common section. This keeps the reference group consistent across each student's questions and avoids pooling the alternative sections.

Access and release. The examination provider permitted the materials to be used for this research. The examination documents and the human-verified structured inputs used by the benchmark are released with the code (`https://anonymous.4open.science/r/Exam2TeachAnonymous-8E75/`) for research use.

D.2 Task-design choices

The teacher discussions described in Section 1 informed the task definitions and quality expectations; they contributed no teacher, school, or classroom records to the benchmark or its evaluation. The document-length requirements set a common output scope across agents and subjects; the two-page worksheet limit also permits printing on one double-sided A4 sheet.

The report's decision cards make the evidence and reasoning behind teaching priorities checkable while leaving agents to choose the diagnoses, classroom checks, and responses. Appendix J provides the full task requirements and the worksheet-selection formula and rationale.

E Evaluation Preparation and Conventions

Section 4 states what each evaluation dimension decides and on what evidence. Appendices F–I give the detailed methods, validation, results, and failure examples for each dimension.

E.1 Origin of the Evaluation Dimensions

This subsection expands the motivation given in Section 4. Our evaluation dimensions come from failures we met while developing the benchmark (Figure 2, lower panels). First trials with frontier models surfaced the most obvious ones: slide text that overlapped, ran off the page, or hid behind

other shapes, and fluent claims the exam data do not support, such as a slide saying most students lost a mark that five of the ten had earned. Expanding to more models introduced omissions of required content, which the strongest models rarely made, such as a deck giving the required question statistics for only some questions. The wider set also exposed feedback that left the student nothing to act on, such as a worksheet that set the student's answer beside the correct one and added a generic tip but never named the error. We therefore evaluate one dimension per failure: *task completion*, *layout quality*, *factual correctness*, and *personalization*, in the order used throughout the paper.

E.2 Data Preparation

This subsection expands Section 4.1. The prepared views retain links between extracted content and rendered pages, so evaluators can locate the evidence behind a block or visual object.

Document delivery and rendering. The report and worksheet prompts require agents to export PDFs from their DOCX files with LibreOffice (The Document Foundation, n.d.); the slide prompt requires a PPTX deck, which is converted to PDF during evaluation. The block extraction pipeline rasterizes the PDFs into PNG page images at 200 dpi, retaining their physical page order. For native DOCX layout diagnostics, the evaluator also makes its own PDF rendering to obtain pagination and geometry (Appendix G.1.1). Task-completion judges render and inspect the delivered files within their own sessions, as described in Section 4.2.

Text-block extraction. GPT-5.5 processes pages in order. Each call receives the current page image and, where available, the preceding page image and the blocks that include that preceding page. The extraction prompt asks for all reliably visible ordinary text in natural reading order, preserving wording, values, qualifiers, and meaning-bearing relationships. Headings and shared labels remain with dependent content when separating them would obscure its meaning. Decorative layout is omitted. The extractor updates an earlier block only when the current page genuinely continues the same semantic unit; a shared topic or repeated heading alone is insufficient. Each block records its index, content, and source pages.

After page extraction, blocks longer than 1,000 characters undergo a separate review for possible splitting. Length triggers review rather than imposing a hard block-size limit: the model splits only at boundaries that preserve independently understandable units and may retain a tightly connected block intact. A deterministic check requires the concatenated parts to reproduce the original content in order, ignoring whitespace. The resulting blocks are reindexed in reading order, with each split part retaining its parent block's source-page list.

Visual-object markers. When an object's meaning depends on its appearance or two-dimensional arrangement, the extractor inserts a typed marker at its reading position and leaves its internal content for image-based inspection. Surrounding ordinary text, such as an independently rendered caption, is transcribed normally. A separate index records each object's type, containing block, marker position within that block, and source pages. The marker and nearby text therefore locate the object without requiring a bounding box. Text whose structure can be represented reliably in plain text remains in the ordinary transcription.

Expanded text blocks. For each block containing markers, a separate GPT-5.5 call receives the block and its page images and returns one description per marker, in marker order. The prompt asks for visible values and meaning-bearing relationships, including table row–column associations and chart scales, legends, and comparisons. Values inferred from visual encodings must be identified as estimates. The script checks the number of descriptions and replaces the markers in a separate copy, preserving ordinary text, block indices, and page lists. Blocks without markers are unchanged. These expanded text blocks supply character counts for the factual correctness score weights (Appendix H.1); factual verification uses the original blocks and images. The extraction, splitting, and expansion prompts, runtime inputs, and validation-retry feedback are provided in our repository.

Task constraints and reference records. Task constraints are stored separately from document content and copied from a task-specific policy into each judge's context. Reference records are constructed by selecting fields from the structured assessment data, matching question identifiers and, for worksheets, the intended student's identifier. This selection does not use the generated

document to decide which evidence to include. Reports and slides receive questions, maximum marks, and cohort statistics; worksheets additionally receive the student's answers and marks with the corresponding marking references. Appendix I.1.2 specifies the fields and exclusions. Each worksheet is paired with only its intended student's pages and records. The prepared document view includes the original blocks, visual-object index, and page images, with expanded text blocks also supplied when available. Validation checks the context schema and the consistency of block indices, page references, and target scope before judging.

E.3 Overall Score

This subsection defines the overall score referred to in Section 4.2 and reported in Tables 1 and 2. For each agent and task t, we average the scores of each dimension across the five subjects with equal weight. Let C_t, L_t, F_t, and P_t denote these average scores for task completion, layout quality, factual correctness, and personalization. The score of task t, S_t, and the overall score (*Total*) of the three tasks are given as follows:

$$S_t = \frac{C_t}{100}\left(0.30\,L_t + 0.40\,F_t + 0.30 \times 12.5\,P_t\right), \qquad \text{Total} = \frac{1}{3}\sum_t S_t, \tag{1}$$

where 12.5 rescales P_t from 0–8 to 0–100 to align with L_t and F_t. We model C_t as a multiplicative factor rather than an additive term because omitting content inflates both layout quality and factual correctness scores, since a shorter output doc tends to contain fewer layout and factual errors. Multiplying by C_t penalizes this behavior by scaling the quality terms by the proportion of required content in the output doc. Among the quality terms, we give F_t the greatest weight (40%) because inaccurate educational content severely undermines teaching and learning, a primary concern voiced by frontline teachers in our feedback sessions. We balance the quality dimensions in (1) to capture an agent's comprehensive capability, while we also evaluate each dimension independently.

E.4 Evaluator Validation

This subsection summarizes the human validation of each evaluator; the per-dimension appendices give the protocols and configuration comparisons. We checked each evaluator against human review of benchmark outputs. These checks test whether an evaluator finds what human reviewers find on the reviewed samples; they do not calibrate the score mappings or pass thresholds (Section 4.2). The samples lean toward Biology and slide decks, so coverage of the other subjects and tasks is thinner.

Task completion. A human reviewer audited all 2,030 checklist items in 30 Biology documents: ten class reports, ten slide decks, and one worksheet per agent. No clearly absent or incomplete required item had received full credit.

Layout quality. Human annotation marked 78 of the 150 pages in five main-experiment slide decks as requiring correction. Under page-level majority voting over three runs, the evaluator flagged 76 pages, 74 of them correctly (94.9% recall, 97.4% precision; 96.1% F1); the benchmark evaluator runs once, and the three runs agreed on 143 of the 150 pages.

Factual correctness. On a 30-page slide deck whose 41 annotated errors span data, knowledge, inference, and chart errors, each of three runs found 39 (95.1% recall). Of 445 errors the evaluator reported on the benchmark's Biology slides, human review confirmed 439 (98.7% precision).

Personalization. Human review classified 60 evaluator criticisms on 18 documents as 33 required issues, 14 optional improvements, and 13 invalid criticisms. The evaluator's three judges together found 31 of the 33 required issues (93.9% recall; an issue counts as found if any judge reported it), and 36 of their 38 reported claims were required or optional (94.7% supported-finding precision).

Appendices F.2, G.2, H.2, and I.2 give the review protocols and the configuration comparisons that informed each evaluator choice.

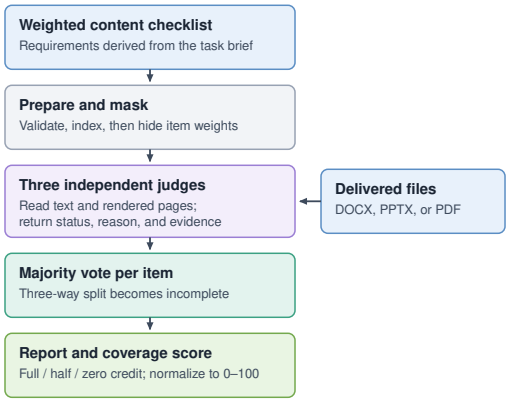


Figure 12: Task completion pipeline: the checklist is masked, three judges inspect the document, and a per-item majority vote sets the credit.

E.5 Evaluation Conventions

Manual output-presence check. Before scoring, we manually checked that the final deliverables existed: one PPTX for each slide deck, a DOCX/PDF pair for each class report, and one DOCX/PDF pair per expected student worksheet (two files per student). Missing deliverables are reported separately from the four dimension scores.

Independence of the dimensions. Taken together, the four dimensions are designed to be mostly independent. Task completion checks whether required content is present, and the other three judge the quality of the content delivered, so task completion neither affects nor depends on them. Two interactions remain among the three quality dimensions. First, layout defects can limit the evidence available to the factual and personalization judges through the rendered-page extraction (Section 4.1). This is intended: the quality dimensions assess documents as their audience receives them. Second, factual correctness and personalization both penalize false claims about student or class performance when those claims guide the material. For example, a worksheet that misquotes a student's answer and explains an error the student never made is penalized under both.

E.6 Failure-Pattern Conventions

The Failure Patterns subsections of Appendices F–I define the failure patterns of each dimension, report their counts in the main experiment, and give illustrative examples. A failure that affects more than one dimension is placed under the dimension responsible for its primary repair and cross-referenced elsewhere only when the second reading matters.

Examples identify the producing agent, subject, task, primary evidence, and a brief explanation of the failure. Textual evidence is quoted or summarized with its source context; task-completion examples cite the affected requirement and document page. Layout quality and visual factual-error examples show the rendered page.

F Task Completion

F.1 Method and Scoring

Each task has one hand-authored checklist, derived from its prompt and used for all five subjects. Its items are independently decidable requirements: 53 for class reports and 111 for slide decks, each of weight 1, and 39 for student worksheets, weighted by importance for a total of 120. Judges receive the items with all weights removed, so an item's value cannot influence its status. Our repository provides the prompts and checklists for all three tasks. Figure 12 shows the panel and scoring steps for the checklist evaluation.

Judge panel. Three judges independently inspect the delivered document file itself, using rendered pages as well as extracted text so that content in charts, images, and decorated shapes is not treated as absent merely because it is difficult to extract. For every judge-scored checklist item, each judge returns exactly one of `present_and_complete`, `present_but_incomplete`, or `absent`, corresponding to *complete*, *incomplete*, and *absent* in Section 4.2, together with a reason and document evidence. Evidence names pages, locations, or quoted text for present content and the places searched for absent content. For worksheets, the judge prompt adds rules that fix how borderline cases map onto the three statuses. For example, the item requiring exactly three focus problems is absent when the worksheet has none, incomplete for any other count, and complete only at three; and a question number or label alone does not count as restating a source question.

A document is scored only when all three judges return valid results: an output that misses an item, or lacks a status, reason, or evidence, is invalid. If any judge's output is invalid, the evaluation fails, and the failed judge is re-run until all three results are valid; we never vote with fewer than three judges.

Vote and scoring. For each item, the final status is the status chosen by a strict majority of the judges. When the panel is split with no majority, the item falls back to `present_but_incomplete`: the panel disagrees between fully present and absent, and the middle status is the natural resolution. Statuses then map onto the item weights: `present_and_complete` earns the full weight, `present_but_incomplete` half, and `absent` nothing. The task completion rate is the share of the total weight earned,

$$\text{Task completion rate} = \frac{\sum_i c_i\, w_i}{\sum_i w_i} \times 100\%, \qquad c_i \in \{1, \tfrac{1}{2}, 0\}, \tag{2}$$

where w_i is the weight of item i and c_i is the credit its voted status earns. Because the checklists differ in size and weighting, rates for different tasks are not on the same scale. For each document, the evaluator records its checklist decisions and the evidence returned by the judges.

Coverage, not correctness. The rate measures whether required content is present, independently of factual correctness or tailoring. A worksheet can achieve 100% task completion while misquoting the student's answer and explaining an error the student did not make, provided every checklist requirement is otherwise covered. Conversely, an explanation that restates the student's answer and the correct one without diagnosing the error may lose only part of one item's credit, because the required explanation is present; whether it diagnoses the error is judged under personalization.

F.2 Validation

Manual audit coverage. We selected 30 Biology documents for manual validation: all ten class reports, all ten slide decks, and one randomly chosen student worksheet per agent. A human reviewer went through every document and inspected all 2,030 checklist items, covering both full-credit and deducted items. The checklists contained 530 report items, 1,110 slide-deck items, and 390 student-worksheet items.

Across all 2,030 items, manual inspection found no case in which a clearly absent or incomplete required item received full credit.

F.3 Result Details

Appendix A reports task completion rates and pass rates. Table 7 instead counts the final status of each checklist item in each document. A report has 53 items, a slide deck 111, and a worksheet 39. Repeated question analyses or focus blocks contribute separate item judgments. Percentages use all item judgments for that task as their denominator. The table therefore counts unmet checklist items; the 2,585 unmet requirements in Section 5.2 group repeated instances of a requirement within each document, as Appendix F.4 describes.

The three tasks use different checklists, item weights, and document counts, so the table does not pool them. The proportion of unmet checklist items is roughly two to three times higher for slides and worksheets than for reports. Appendix F.4 groups unmet items by requirement and gives examples of the most common ones.

Table 7: Task-completion checklist item statuses by document task. Counts are item judgments, not unique documents. Percentages use the judgments in each row; unmet items combine absent and incomplete statuses.

Task	Documents	Items judged	Absent	Incomplete	Unmet items
Reports	50	2,650	39 (1.5%)	131 (4.9%)	170 (6.4%)
Slides	50	5,550	241 (4.3%)	588 (10.6%)	829 (14.9%)
Worksheets	600	23,400	1,109 (4.7%)	3,419 (14.6%)	4,528 (19.4%)

F.4 Failure Patterns

We audit unmet requirements by document task: class reports, slide decks, and student worksheets. The task requirements are summarized in Section 2. Counts use the voted checklist results for the five-subject main experiment. Repeated slide analyses and worksheet blocks are grouped by requirement; a document is counted once when any instance is absent or incomplete. Requirements can overlap within a document, so their counts should not be added. At least one checklist requirement is absent or incomplete in 47/50 reports, 46/50 slide decks, and 509/600 worksheets. A document can have an unmet requirement and still pass the 95% task-completion threshold.

F.4.1 Class reports

Each report ranks class learning priorities and, for the top two, sets out a classroom check that separates competing causes, an initial teaching response, and what to do next depending on the outcome. The most common unmet requirement is incomplete evidence for cited questions: 45/50 reports omit the expected answer or marking point for at least one question used to support or challenge a diagnosis or teaching decision. Other recurrent unmet requirements are an unspecified method for collecting every learner's diagnostic response (19/50) and no next action when the first teaching response meets its success criterion (16/50). In Grok 4.5's Biology report, Q10(c) is used to justify deprioritization with a score rate and common wrong-answer pattern, but its expected answer is not given (p. 3). Marking expectations for the main priorities do not complete the evidence for this additional decision.

F.4.2 Slide decks

Each deck presents an exam-performance overview, analyzes five selected questions, and ends with a class review plan. Question-level attempt rates are absent or incomplete in 28/50 decks. Across the five required question analyses, at least one blank-answer rate is absent or incomplete in 26/50 decks, and marks lost for missed scoring points in 24/50. GPT-5.5's Chemistry deck states attempt rates for selected questions, including Q11 and Q9, but its overview does not cover the remaining questions (slides 9, 20, 23, and 29). An attempt rate measures whether a learner answered; a zero-mark answer can still be an attempt.

F.4.3 Student worksheets

Each learner receives a score overview, three source-question focus blocks explaining lost marks, and one matched practice problem per block, with answers collected upside down in a final answer key. At least one of the three focus blocks lacks a complete explanation of why the learner's answer lost marks in 402/600 worksheets. At least one source question is not fully restated in 395/600, and the required upside-down answer key is absent or incomplete in 234/600. In GLM 5.2's Physics worksheet for Student 006, the Q11(b)(i) block shows the learner's incorrect current calculation and supplies the correct formula, but does not explain the learner's error (p. 1). Providing a correct method does not fulfill the requested explanation of lost credit.

G Layout Quality

G.1 Method and Scoring

Defect categories. The four categories in Section 4.2 are named **off-page clipping**, **text overlap** (text with text or with an image), **container crossing or occlusion**, and **legibility** below. Container

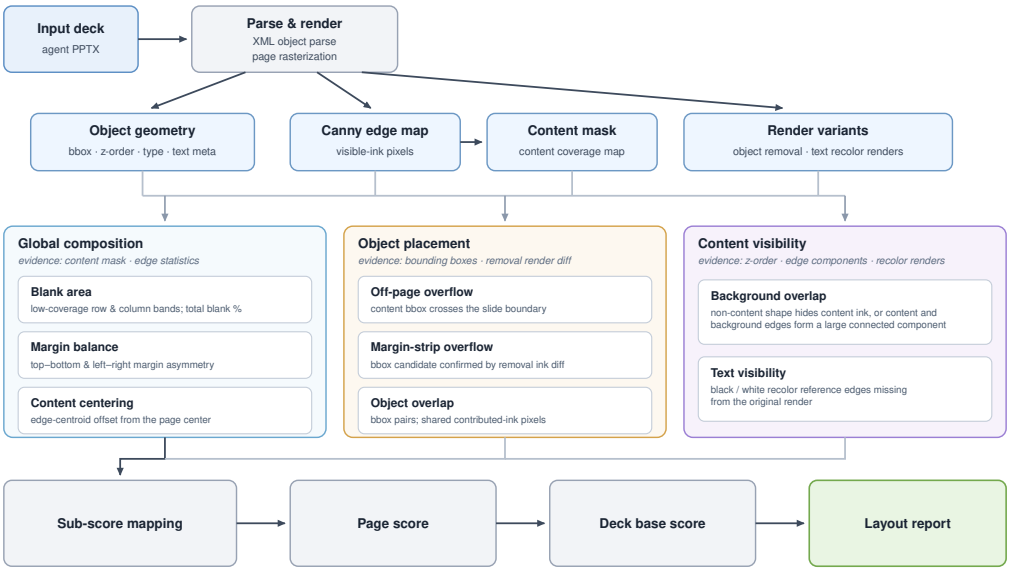


Figure 13: Logic of the PPTX checker script released in our repository. Top to bottom: the deck is parsed and rendered; four kinds of evidence are derived; three groups of checks read them; the bottom row computes the native deck score. The flags sent to the judge come from the object placement and content visibility checks.

crossing includes text extending beyond its card or table cell; an intentional background behind its own text is not an occlusion. Legibility includes low contrast, excessively small type, malformed mathematical notation, and dense formatting that impedes scanning even when individual characters remain recognizable. Normal scientific symbols and readable plain-text notation are valid, and text volume or proximity alone does not warrant a penalty. Factual correctness and pedagogical adequacy fall outside this evaluator.

The evaluation has three steps. Scripts screen the document for candidate defects and record each as a flag with its page position and object identifier; a judge inspects every page image, with the flags where they exist; and the confirmed findings are scored. The two scripts, one for PPTX and one for DOCX, are provided in our repository. Each also computes its own layout quality score, which we call the native score; the benchmark uses only the judge's confirmed findings, and the native PPTX score serves only as the fixed-script comparison in Table 9.

G.1.1 Script Screening

Figure 13 shows the logic of the PPTX script. The checker normalizes the deck through LibreOffice, parses the slide objects, and rasterizes each page. From these it derives four kinds of evidence: the geometry of every text, image, and decorative shape, with bounding boxes and stacking order; an edge map of visible ink; a content-coverage mask; and controlled renders that remove an object or recolor text, which separate visible interference from merely intersecting boxes. Three groups of checks read this evidence. Global composition measures blank area, margin balance, and content centering. Object placement detects off-page overflow, margin-strip overflow, and object overlap. Content visibility detects a background shape hiding content and text lost against its background. The flags passed to the judge come from the object placement and content visibility checks, which correspond to off-page clipping, text overlap, and container crossing or occlusion; legibility is judged visually only. The bottom row of the figure is the native score: each check maps to a sub-score, a page score is their weighted mean, and the deck score applies a nonlinear mean per check and subtracts systemic penalties.

For DOCX reports and worksheets, the checker first paginates the editable file with LibreOffice and extracts geometry and visible ink from that checker-made PDF, rather than the separately submitted export. Text boxes can include font-metric slack, so rendered ink is used to confirm geometric

candidates. Checks cover page fill, edge and off-page overflow, text and float collisions, background interference, visibility, and clipping. Controlled renders recolor text or remove non-wrapping floats; if a controlled render shifts glyph positions, the checks that depend on it are marked degraded rather than trusted. Unlike PPTX, the DOCX checker omits margin balance and content centering because flowing pages are normally top-anchored and asymmetric margins can be intentional.

G.1.2 VISUAL INSPECTION

The judge inspects every page image. A page without script flags receives an image-only, full-page inspection. A page with flags is inspected with the flags: the judge confirms or rejects each one and also searches the rest of the page for defects the scripts missed. If the judge clears a flagged page, the page is inspected once more with neither the flags nor the first verdict, so that the second pass is not anchored on either; findings from either pass are kept. Visual inspection also finds defects that geometry cannot, such as unreadable notation and excessive scanning burden.

The judge is Gemini 3.7 Flash at temperature zero, asked for a binary defect/no-defect decision and localized evidence per page. Each slide is inspected on its own; the pages of a report or worksheet are supplied together, and findings are attributed to individual pages. One evaluator prompt, provided in our repository, serves all three tasks; only this input grouping follows the document format.

G.1.3 SCORING

Confirmed findings are deduplicated by category and affected object; overlap is counted by distinct object pair. Where no script flag identifies the object, an object is an identifiable visual block, cell, or continuous dense region; a paragraph is not split into lines to multiply counts. Formula-rendering and density problems count under legibility rather than receiving separate weights. The scoring retains the count-based sub-scores, nonlinear means, and systemic penalties of the native PPTX score, applied to the confirmed findings in the four evaluator-confirmable categories with renormalized weights (Table 8). Native blank-page, margin, centering, and overflow-margin components have zero weight in this adaptation, and its blank-page penalty is disabled.

Table 8: Category weights and page-level scores for confirmed layout defects.

Category	Normalized weight	Category score for n confirmed defects on a page
Off-page clipping	22/78	100, 25, 10, 0 for n = 0, 1, 2, $\geq$3
Text overlap	20/78	100, 25, 10, 0 for n = 0, 1, 2, $\geq$3
Container crossing or occlusion	18/78	max(0, $100 - 25n$)
Legibility	18/78	max(0, $100 - 25n$)

Document aggregation accounts for both defect counts and recurrence. Let m_c denote the mean page-level category score over all pages, r_c the fraction of pages with at least one defect, and u_c the mean number of defects per page over all pages, in category c. With categories ordered as in Table 8, the document score is

$$S = \mathrm{clip}_{[0,100]}\left[\sum_{c=1}^{4} w_c\, 100(m_c/100)^{\gamma_c} - P_1 - P_2 - P_{34}\right], \tag{3}$$

where $w = (22, 20, 18, 18)/78$, $\gamma = (1.15, 1.35, 1.35, 1.35)$, $P_c = \min(16, 36r_c + 9u_c)$ for $c \in \{1, 2\}$, and

$$P_{34} = \min\left(8, \max\left(0, \frac{8((r_3 + r_4)/2 - 0.05)}{0.95}\right)\right).$$

An isolated defect therefore does not invalidate an entire page, while recurrent failures incur additional deductions. The validation in Appendix G.2 tests whether the evaluator detects pages that contain defects; the penalty magnitudes are fixed design choices.

G.2 VALIDATION

Human reference. We validate page-level detection on complete slide decks rather than on flagged pages only, so that both false alarms and missed defects are counted. The reference covers five 30-page decks from the main experiment: Biology/Grok 4.5, Chemistry/Kimi K2.6, English reading/MiMo V2.5 Pro, Mathematics/MiniMax M3, and Physics/Claude Opus 5. The reference contains no report or worksheet pages, so the DOCX path is validated only through the judge and prompt it shares with slides.

Annotators first labeled every page without seeing any evaluator output. Selected disagreements were then re-examined with the script and judge evidence visible, and three reviewers independently confirmed the resulting labels. Two later reviews, also with evidence visible, re-examined 18 pages on which the adopted evaluator had erred: four with excess false positives and 14 on which it erred in at least one of its three runs. The final reference adopts the labels from these later reviews, which were not re-confirmed by all three reviewers. A positive label requires at least one defect warranting a deduction; pages with only optional improvements count as negatives. The final reference has **78 positive pages and 72 negative pages**, 34 without issues and 38 with optional improvements. All four detectors below are scored against this same final reference.

Detectors compared. We compare the full native script, standalone visual inspection of single pages, and the adopted evaluator, which inspects pages with script flags and gives a cleared flagged page a second pass. Both visual detectors use the same core prompt, Gemini 3.7 Flash at temperature zero, the same page images, output schema, and bounded format-repair rules, and three complete runs each. Standalone inspection receives no script flags or object anchors and has no second pass. To isolate the second pass, we also score the adopted evaluator's first-pass predictions, taken from the same runs before any second pass.

Majority voting. Visual detectors are scored by page-level majority voting across their three runs: a page is positive if at least two runs report a required defect. Let $d_{p,r} \in \{0, 1\}$ denote the prediction for page p in run r. The aggregated decision is

$$\widehat{d}_p = \mathbf{1}\left[\sum_{r=1}^{3} d_{p,r} \geq 2\right]. \tag{4}$$

Precision (the share of flagged pages that are positive), recall (the share of positive pages flagged), and F1 are computed once from the aggregated decisions, not averaged over runs. The native script is deterministic and needs no voting. Voting is used only for this validation; in the benchmark, the evaluator runs once per document.

Table 9: Page-level defect detection on the 150-page five-subject reference. Visual rows use majority voting across three runs; the native script has fixed predictions. Optional improvements count as negatives.

Detector	Precision (%)	Recall (%)	F1 (%)
Full native script	72.92	89.74	80.46
Standalone visual inspection, same prompt	98.61	91.03	94.67
Inspection with script flags, without second pass	98.63	92.31	95.36
Inspection with script flags, with second pass (adopted)	**97.37**	**94.87**	**96.10**

Findings. Table 9 supports three conclusions. First, visual inspection is what removes the script's false alarms. The full script reaches 89.74% recall but only 72.92% precision, flagging 26 negative pages; every visual detector improves recall and reduces false alarms to one or two pages. Second, script flags alone make little difference under the same prompt. First-pass inspection with flags finds one additional positive page relative to standalone inspection, with one false alarm for each; this small difference does not establish that the flags improve page decisions. Third, the second pass is what raises recall. Each of the three runs gives 11, 12, and 11 pages a second pass, adding 1, 3, and 2 true positives against 1, 1, and 0 false positives; after voting, the adopted evaluator finds 74 of the 78 positive pages with two false alarms, compared with 71 and one for standalone inspection. We adopt the second pass for this recall: in the benchmark, a missed defect leaves the page unpenalized, whereas a false alarm costs at most one deduction on one page.

The adopted evaluator's final predictions agree across all three runs on 143 of the 150 pages (95.33%). Three positive pages are missed in every run, and no negative page is flagged in all three runs.

G.3 Failure Patterns

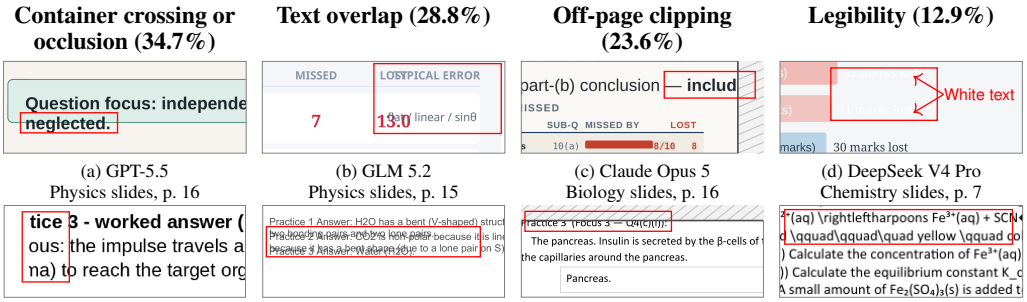
This section covers rendering and organization failures that obscure content or materially burden reading in the delivered document.

G.3.1 Classification and Coverage

The catalog uses the four defect categories of the layout quality evaluator as its families. A case is placed under the family that best describes its primary reading consequence; a single rendered defect can also illustrate a second family when it crosses both a card boundary and the slide edge (as in the chemistry slide in Patterns 1.1 and 3.1). The catalog has four families and eight patterns, each with at least one example.

G.3.2 Pattern Catalog

The catalog follows the order of the four categories in Appendix G.1. Each example identifies the agent, task, subject, and page, and shows the rendered page or the relevant region of it. The English-writing example is outside the five-subject main experiment. Red boxes mark the objects at issue. Slides are delivered as PPTX and shown as rendered through LibreOffice for evaluation; reports and student worksheets are shown as the delivered PDFs.



(a) GPT-5.5 Physics slides, p. 16
(b) GLM 5.2 Physics slides, p. 15
(c) Claude Opus 5 Biology slides, p. 16
(d) DeepSeek V4 Pro Chemistry slides, p. 7
(e) GLM 5.2 Biology worksheet student 003, p. 2
(f) Qwen 3.7 Max Chemistry worksheet student 001, p. 2
(g) MiniMax M3 Biology worksheet student 002, p. 2
(h) MiniMax M3 Chemistry worksheet student 008, p. 1

Figure 14: Cropped examples of the four layout-defect categories, ordered by their share of 2,274 layout findings. The top row shows PPTX slides; the bottom row shows DOCX worksheets. Red annotations mark the affected regions.

G.3.3 Family 1: Off-page clipping

Content is laid out beyond the page boundary and the overflowing part is cut away. The loss is silent: the sentence ends without any mark of truncation, so a reader cannot tell that text is missing.

1.1 Text cut at the page edge. A text object extends past the page or slide boundary. The visible fragment ends mid-word or mid-phrase.

Example (Slides; Biology; Claude Opus 5; p. 16). The subtitle and all three lines of the lower summary panel run past the right slide edge and are cut. The evaluator records four off-page objects on this page, and the same right-edge clipping recurs on 27 of the deck's 30 slides, which points to a fixed template rather than an isolated overflow.

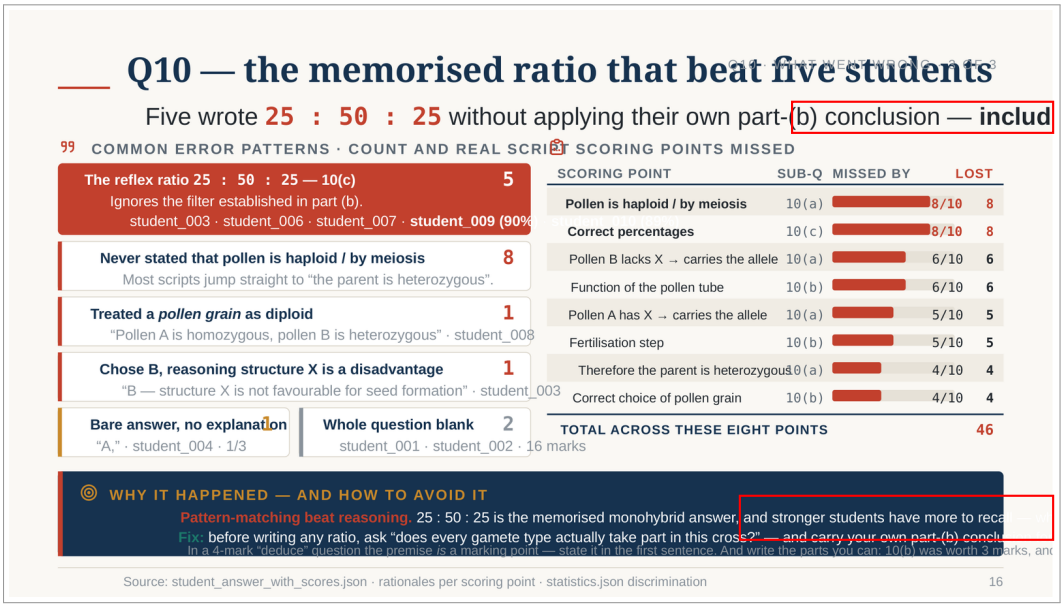


Figure 15: Pattern 1.1: Claude Opus 5, Biology slides, p. 16. The subtitle and the panel lines are cut at the right edge.

Example (Slides; Chemistry; GPT-5.5; p. 30). All four review steps state a success check in the right-hand column, and all four are cut at the slide edge: "Check: 8/10 score at least 5/9 on a similar" is cut mid-sentence. The left column stays inside the page, so what the slide loses is exactly the criterion that makes each step verifiable.

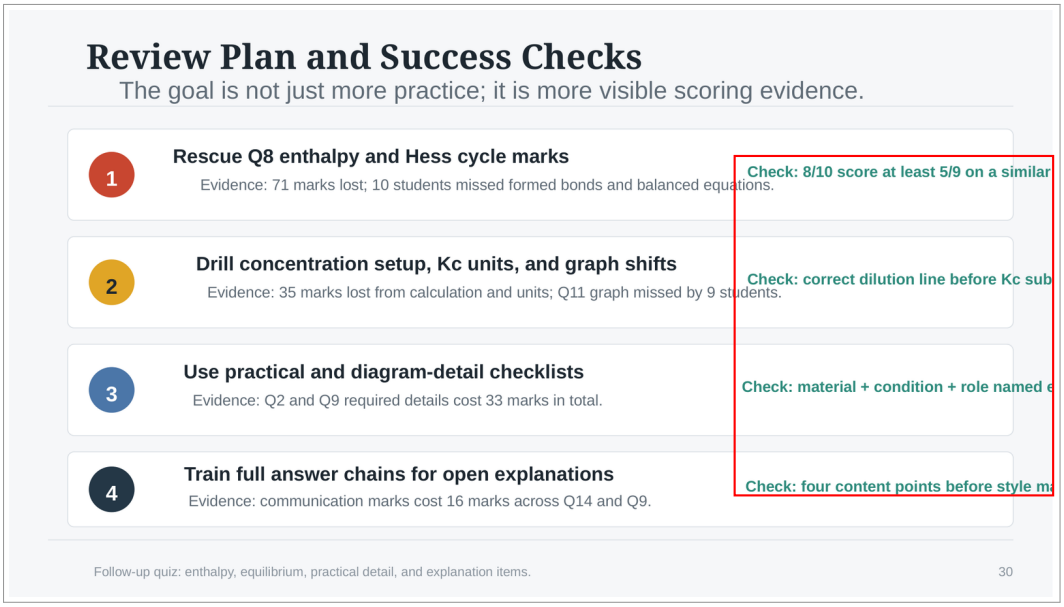


Figure 16: Pattern 1.1: GPT-5.5, Chemistry slides, p. 30. The success check of every review step is cut at the right edge.

G.3.4 Family 2: Text–text or text–image overlap

Two independently positioned objects occupy the same region and print on top of each other. Unlike clipping at a page edge, overlap can obscure content in either or both objects.

2.1 Heading and body collisions. A title, subtitle, or section marker overprints adjacent text, typically after the title wraps to an additional line that the layout does not reserve space for.

Example (Slides; English writing; GPT-5.5; p. 9). The title wraps to a second line, and "solutions" prints across the subtitle "Only 2 of 5 scripts covered ways to overcome strongly." Two coverage labels below repeat the pattern: "challenges" overprints "3/5 covered; 16 estimated marks lost". The overlap follows content length, since the same template is legible where the title fits on one line.

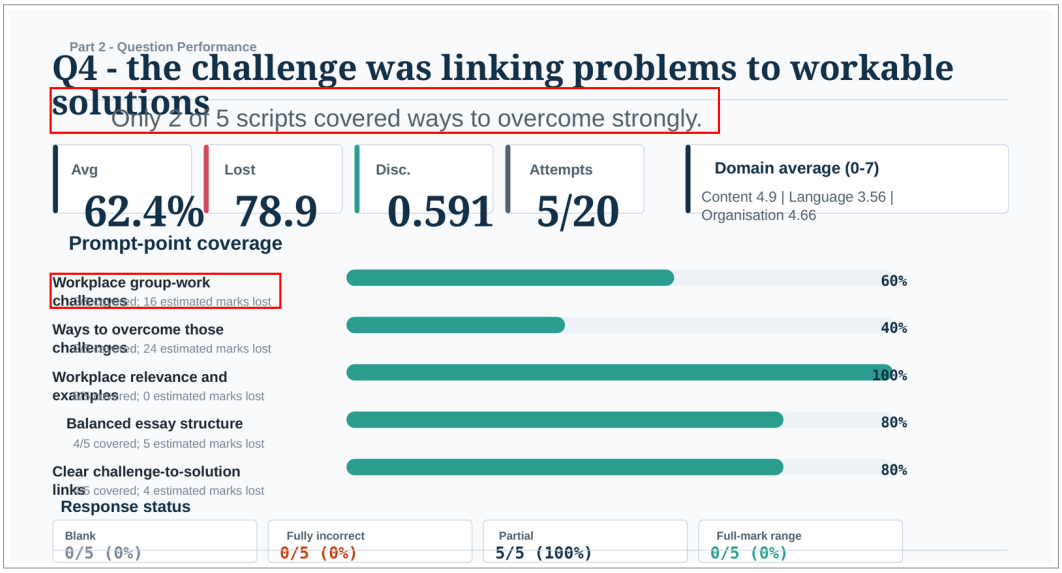


Figure 17: Pattern 2.1: GPT-5.5, English writing slides, p. 9. The wrapped title prints over the subtitle, and the coverage labels over their annotations.

2.2 Title/header collision with chart-label interference. The page combines a heading collision with a separate chart-label placement failure. The heading collision obscures text; the chart quantities remain distinct but are difficult to parse at their shared anchor.

Example (Slides; English reading; Claude Opus 5; p. 9). The title "View B — ranked by total marks lost" overprints the header label "PART 1 · OVERALL PERFORMANCE". On the same page, every bar carries its total-marks value and its error rate at one anchor, so the top bar reads "5858% err" for a value of 58 and a rate of 58%, and the next reads "5344% err". The merged labels force readers to disentangle the value and rate at each bar.

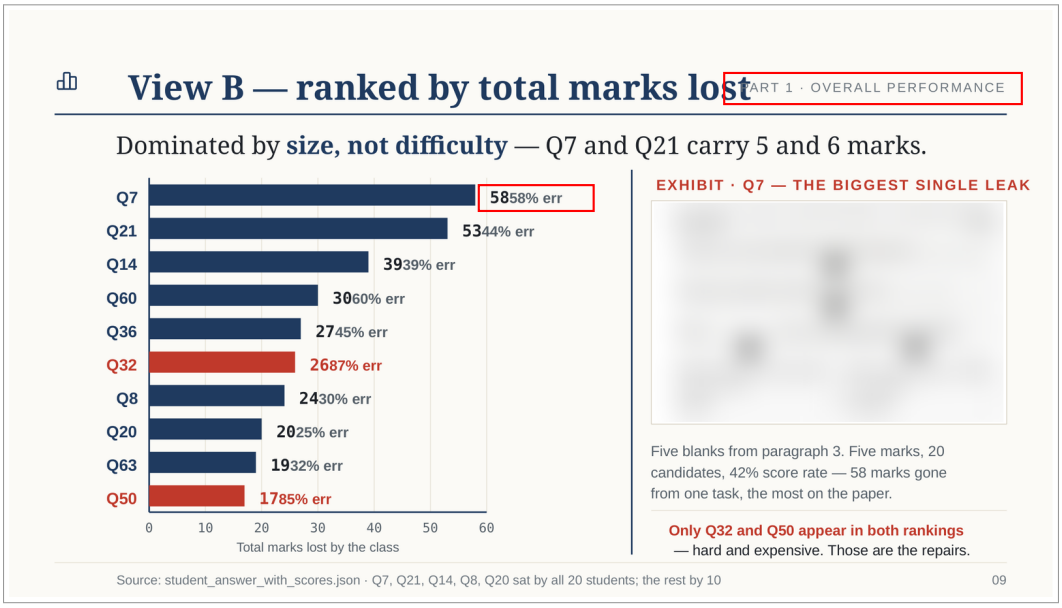


Figure 18: Pattern 2.2: Claude Opus 5, English reading slides, p. 9. The title meets the section marker; value and error-rate labels share an anchor, making the two quantities difficult to distinguish. The source question crop is blurred.

G.3.5 Family 3: Container crossing or occlusion

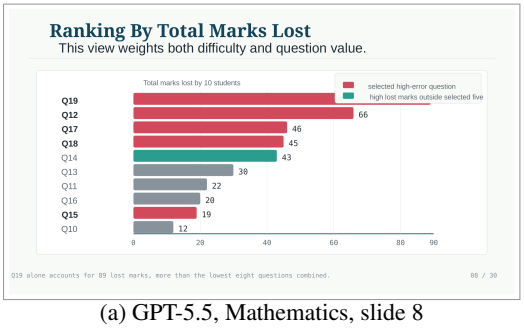
Content stays inside the page but leaves the card, cell, or panel that is meant to hold it, or covers a neighboring object. The page remains complete; the defect is in how space is allotted within the page, not at its edge.

3.1 Text leaving its container. Text extends beyond the visible boundary of its card or table cell, breaking the grouping the container expresses.

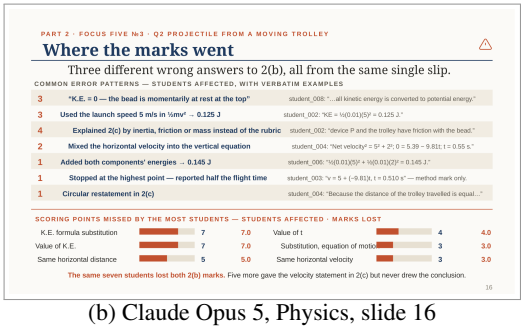
Example (Slides; Chemistry; GPT-5.5; p. 30). On the page shown in Figure 16, each success check also crosses the right boundary of its own white card before reaching the slide edge, breaking the grouping the card expresses.

3.2 Occlusion of a neighboring value. An object is placed over a region another object already uses, hiding information that is itself correctly rendered.

Examples (Slides; Mathematics; GPT-5.5; p. 8; and Physics; Claude Opus 5; p. 16). In the Mathematics deck, a chart legend covers the highest-ranked bar's value (Figure 19a). In the Physics deck, a long skill label extends beneath an adjacent bar (Figure 19b).



(a) GPT-5.5, Mathematics, slide 8



(b) Claude Opus 5, Physics, slide 16

Figure 19: Pattern 3.2: neighboring objects obscure chart information. In (a), the legend covers the highest-ranked bar's value. In (b), the second label in the lower-right bar chart extends beneath its bar.

G.3.6 FAMILY 4: LEGIBILITY

Characters are positioned correctly, yet the page cannot be read as intended. This family covers unrendered notation, density that defeats scanning, and type that is too small or too low in contrast to resolve.

4.1 Unrendered formula markup. Source markup reaches the reader in place of the notation it encodes: LaTeX commands, delimiters, or Markdown syntax appear as literal characters.

Example (Student worksheet; Physics; GPT-5.5; p. 1). The reproduced student answer reads "By F_cen = fracmv^2r", with the fraction command flattened into the text, and a later item ends at "The new period would be frac1". The formula that the diagnosis turns on is delivered as its own markup, and the second fragment stops mid-expression. Six of this agent's ten Physics worksheets carry the same defect.

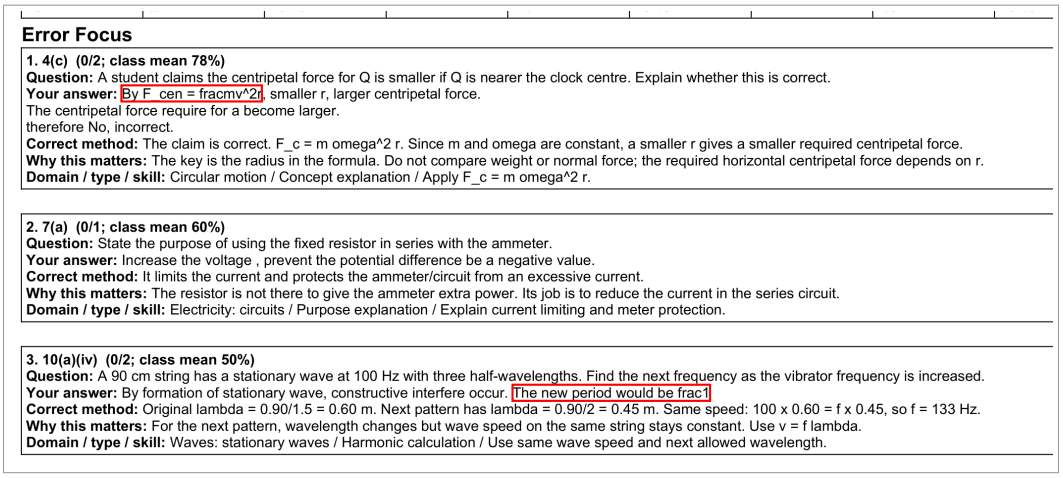
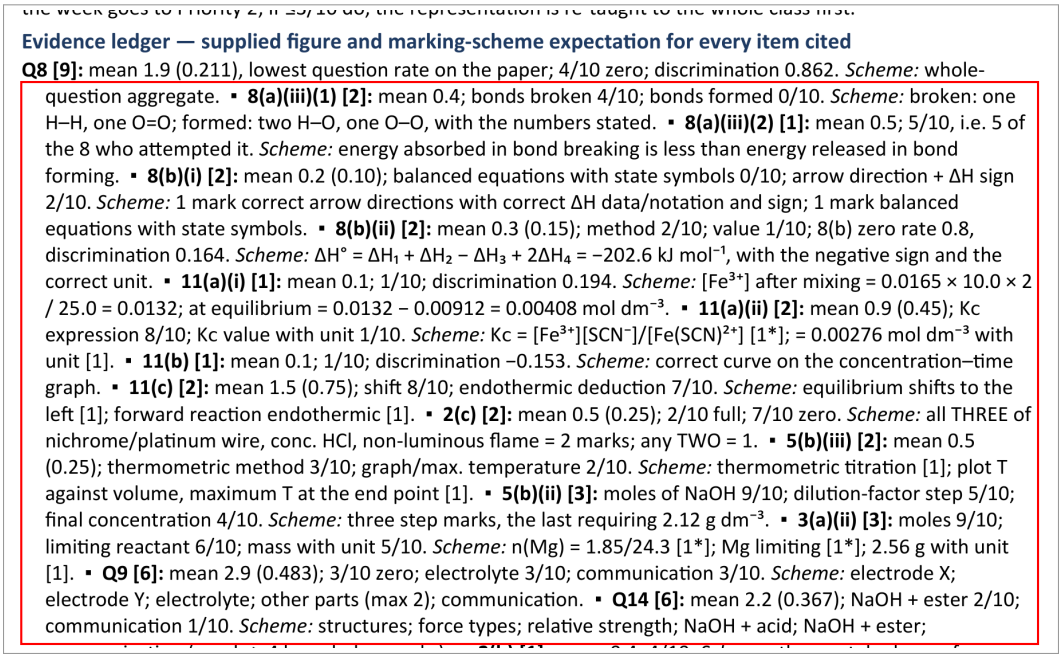


Figure 20: Pattern 4.1: GPT-5.5, Physics student worksheet, p. 1 (Student 005). Fraction commands appear as literal text.

4.2 Density and spacing. Content is legible character by character but is packed into an undifferentiated run, so locating any single item requires reading the whole block.

Example (Report; Chemistry; Claude Opus 5; p. 1). The evidence ledger merges more than fifteen question-level diagnoses and their marking-scheme expectations into one continuous paragraph, separated only by inline bold item labels and middle dots. The information a teacher needs per question is present and correct; the page supplies no structure by which to reach one item of it.



Evidence ledger — supplied figure and marking-scheme expectation for every item cited

Q8 [9]: mean 1.9 (0.211), lowest question rate on the paper; 4/10 zero; discrimination 0.862. *Scheme:* whole-question aggregate. ▪ **8(a)(iii)(1) [2]:** mean 0.4; bonds broken 4/10; bonds formed 0/10. *Scheme:* broken: one H–H, one O=O; formed: two H–O, one O–O, with the numbers stated. ▪ **8(a)(iii)(2) [1]:** mean 0.5; 5/10, i.e. 5 of the 8 who attempted it. *Scheme:* energy absorbed in bond breaking is less than energy released in bond forming. ▪ **8(b)(i) [2]:** mean 0.2 (0.10); balanced equations with state symbols 0/10; arrow direction + ΔH sign 2/10. *Scheme:* 1 mark correct arrow directions with correct ΔH data/notation and sign; 1 mark balanced equations with state symbols. ▪ **8(b)(ii) [2]:** mean 0.3 (0.15); method 2/10; value 1/10; 8(b) zero rate 0.8, discrimination 0.164. *Scheme:* $\Delta H^\circ = \Delta H_1 + \Delta H_2 - \Delta H_3 + 2\Delta H_4 = -202.6$ kJ mol^{-1}, with the negative sign and the correct unit. ▪ **11(a)(i) [1]:** mean 0.1; 1/10; discrimination 0.194. *Scheme:* $[Fe^{3+}]$ after mixing = 0.0165 × 10.0 × 2 / 25.0 = 0.0132; at equilibrium = 0.0132 − 0.00912 = 0.00408 mol dm^{-3}. ▪ **11(a)(ii) [2]:** mean 0.9 (0.45); Kc expression 8/10; Kc value with unit 1/10. *Scheme:* Kc = $[Fe^{3+}][SCN^-]/[Fe(SCN)^{2+}]$ [1*]; = 0.00276 mol dm^{-3} with unit [1]. ▪ **11(b) [1]:** mean 0.1; 1/10; discrimination −0.153. *Scheme:* correct curve on the concentration–time graph. ▪ **11(c) [2]:** mean 1.5 (0.75); shift 8/10; endothermic deduction 7/10. *Scheme:* equilibrium shifts to the left [1]; forward reaction endothermic [1]. ▪ **2(c) [2]:** mean 0.5 (0.25); 2/10 full; 7/10 zero. *Scheme:* all THREE of nichrome/platinum wire, conc. HCl, non-luminous flame = 2 marks; any TWO = 1. ▪ **5(b)(iii) [2]:** mean 0.5 (0.25); thermometric method 3/10; graph/max. temperature 2/10. *Scheme:* thermometric titration [1]; plot T against volume, maximum T at the end point [1]. ▪ **5(b)(ii) [3]:** moles of NaOH 9/10; dilution-factor step 5/10; final concentration 4/10. *Scheme:* three step marks, the last requiring 2.12 g dm^{-3}. ▪ **3(a)(ii) [3]:** moles 9/10; limiting reactant 6/10; mass with unit 5/10. *Scheme:* n(Mg) = 1.85/24.3 [1*]; Mg limiting [1*]; 2.56 g with unit [1]. ▪ **Q9 [6]:** mean 2.9 (0.483); 3/10 zero; electrolyte 3/10; communication 3/10. *Scheme:* electrode X; electrode Y; electrolyte; other parts (max 2); communication. ▪ **Q14 [6]:** mean 2.2 (0.367); NaOH + ester 2/10; communication 1/10. *Scheme:* structures; force types; relative strength; NaOH + acid; NaOH + ester;

Figure 21: Pattern 4.2: Claude Opus 5, Chemistry report, p. 1. The evidence ledger runs as a single block.

4.3 Size and contrast. Type is rendered too small, at too low a resolution, or with too little contrast against its background to be resolved at normal reading size.

H Factual Correctness

H.1 Method and Scoring

H.1.1 Pipeline Stages

Figure 22 summarizes the verification and review stages.

Batching. The prepared text blocks and visual objects (Section 4.1) form two ordered batch streams: text batches contain at most 10,000 characters of block text, and visual batches at most five objects. Verification reads the original blocks and page images; the expanded text blocks are used only for scoring.

Parallel textual and visual verification. Verifiers process the text and visual batches in parallel. Text verifiers check claims introduced by ordinary text, reading bounded relevant context and inspecting images when needed. Visual verifiers inspect the assigned objects, checking explicit content and facts encoded by position, size, color, grouping, or connection, with appropriate tolerance for visually estimated values. Both compare source-related claims against the supplied exam data, performing necessary calculations and reasoning; claims unrelated to the exam data are judged from the document's premises, valid logic, and reliable common knowledge. Verifiers report only `incorrect` or `unverifiable` claims, with a specific error and supported correction for each incorrect finding.

Output check, independent review, and consolidation. Each batch result passes a deterministic output check and then independent review. The reviewer rechecks the findings against the same evidence, removes or repairs mistaken items, and searches the assigned material for omissions. Reviewed results are checked again and consolidated across both streams. Our repository provides the verifier and reviewer prompts.

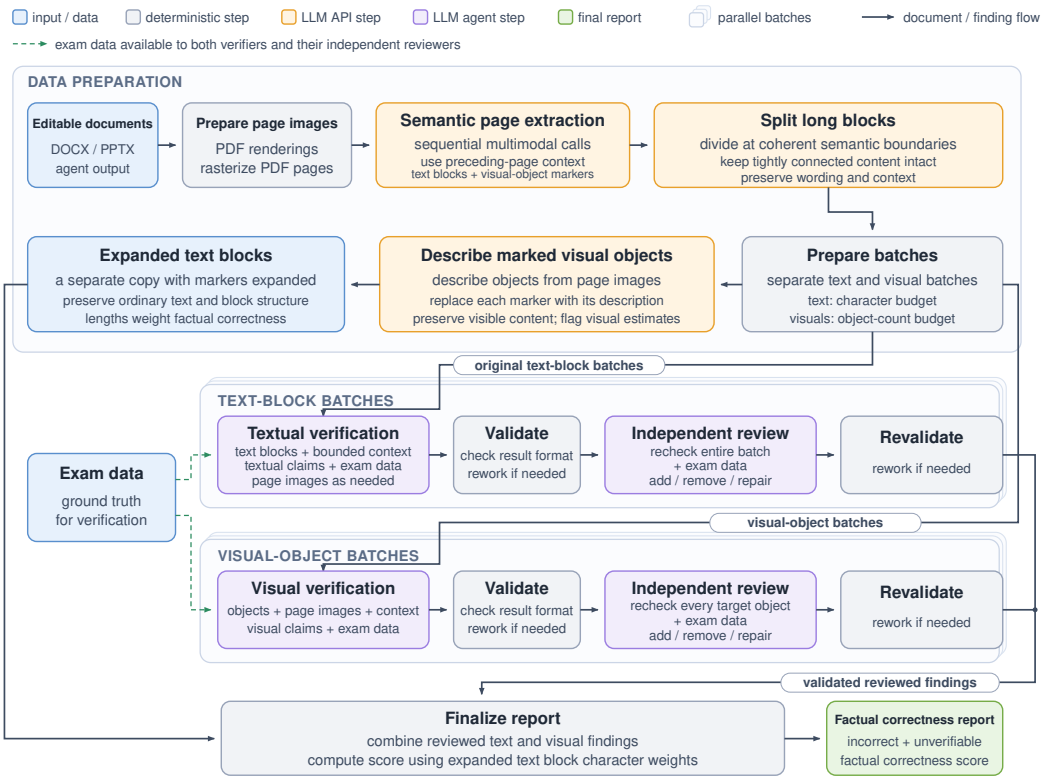


Figure 22: Factual correctness evaluation pipeline. Data preparation (Section 4.1) supplies text blocks, visual-object markers, page images, and expanded text blocks. Textual and visual batches undergo verification and independent review before findings are consolidated and scored.

Scoring. Let the size of a block be the character count of its expanded text block (Appendix E.2), so visual objects count through their inserted descriptions. After consolidation, each block is classed as erroneous if it contains at least one confirmed incorrect claim, unverifiable if it contains unverifiable claims and no confirmed error, and clean otherwise; a block is counted once regardless of the number of findings within it. With A the total size of erroneous blocks, B the total size of unverifiable blocks, and C the total size of all blocks, the factual correctness score is

$$100 \times \left(1 - \frac{A + 0.5B}{C}\right). \tag{5}$$

H.2 Validation

To test whether the evaluator can identify factual errors, we selected one agent-generated 30-page slide deck and annotated a reference set of 41 factual errors. The set spans data errors, errors of factual knowledge, faulty inference or summarization, and errors conveyed inside charts; detecting them requires checking calculations, statistical claims, analyses, and summaries. Each study below runs an evaluator variant on the same deck, manually matches its factual-correctness report to the reference set, and calculates recall as the fraction of the 41 reference errors detected.

The evaluator targets consequential factual errors, which we define as materially wrong content: incorrect data or numerical results, errors of factual knowledge, invalid calculations or conclusions, and clear contradictions with the supplied exam data. Minor imprecision in wording, qualification, or rigor is outside this reference definition, and borderline rigor or wording findings are not counted as detected reference errors. Clear false positives were too rare in this experiment to compare precision across variants, so this study reports recall; Appendix H.2.4 measures precision on benchmark outputs.

Reporting protocol. Each variant has three runs. We report the number of reference errors found in each run and the mean recall, $\bar{R} = (r_1 + r_2 + r_3)/(3N)$, where $N = 10$ for the first 10 pages and $N = 41$ for the complete deck.

H.2.1 Direct Verification by a Single Model

Experimental design. This experiment tests how well a single harness–model pair, used directly as the evaluator, identifies factual errors, and whether that ability changes as the amount of material grows. We evaluate the seven candidates listed in Table 10 under two conditions. In the *10-page* condition, the candidate receives only the first 10 pages of the deck and is scored against the 10 reference errors on those pages. In the *30-page* condition, it receives the complete deck and is scored against all 41 reference errors. In both conditions the candidate performs the same factual-correctness task with the slides and exam data, and its report is matched manually to the applicable reference set.

Table 10: Recall of direct verification by a single model on the first 10 pages and on the complete 30-page deck. "Found" lists the three run counts; recall is their mean divided by the reference-set size N. Bold marks the highest mean recall within each condition.

Evaluator (harness / model)	**10 pages** ($N = 10$)		**30 pages** ($N = 41$)	
	Found	**Mean recall**	**Found**	**Mean recall**
Codex / GPT-5.5	10, 10, 8	**93.3%**	30, 28, 30	71.5%
Claude Code / Claude Opus 5	9, 8, 9	86.7%	34, 34, 33	**82.1%**
OpenCode / MiniMax M3	3, 2, 4	30.0%	16, 12, 20	39.0%
OpenCode / Qwen 3.7 Plus	5, 4, 1	33.3%	5, 16, 9	24.4%
OpenCode / Kimi K2.6	4, 1, 5	33.3%	17, 15, 9	33.3%
OpenCode / Gemini 3.5 Flash	4, 5, 4	43.3%	16, 25, 22	51.2%
OpenCode / GLM-5V-Turbo	2, 1, 1	13.3%	3, 1, 1	4.1%

Results and interpretation. GPT-5.5 has the highest mean recall on the first 10 pages (93.3%), detecting all 10 errors in two runs, but falls to 71.5% on the complete deck, a 21.8 percentage-point decrease. Claude Opus 5 leads on the complete deck at 82.1%, down from 86.7% on the first 10 pages. The other candidates range from 4.1% to 51.2% on the complete deck, and some vary substantially across runs: Qwen detects between 5 and 16 errors, and Gemini between 16 and 25. Direct verification thus leaves substantial gaps even for the strongest model on this deck. The effect of the longer input is not uniform: Gemini and MiniMax improve in mean recall on the complete deck, while Kimi is unchanged. Because the longer condition also adds 31 reference errors, the comparison does not separate input length from error difficulty; the following experiments therefore hold the 30-page reference set fixed.

H.2.2 Comparison and Selection of Task-Decomposition Strategies

Experimental design. This experiment compares methods that control the verifier's workload by decomposing the evaluation task. Every variant is run on the complete 30-page deck and evaluated against the same 41-error reference set. All variants use Codex with GPT-5.5 as the verifier, and the supplied exam data, manual matching protocol, and recall definition are held fixed, so the comparison isolates the decomposition strategy.

Vertical decomposition. The vertical pipelines separate construction of the verification subject from its final factual verification, while keeping that subject complete rather than batching it. At the extraction stage, the `direct` baseline uses the original document without an intermediate representation. `content` and `blocks` also start from the original document but extract checked page-wise content and text blocks, respectively. `claims` and `questions` take the `content` result as input and further extract atomic claims or verification questions.

At the verification stage, one final verifier receives the complete original document, content extraction, claim set, question set, or block set. In each base strategy—including the `direct` baseline—that verifier searches the full exam data itself. A corresponding `_groundtruth` strategy instead

assigns retrieval from the exam data to a separate call and gives the final verifier only the checked, task-relevant excerpts. Vertical strategies therefore control workload by offloading subject extraction or exam-data retrieval, not by distributing the final decisions among multiple verifiers.

Horizontal decomposition. The horizontal pipelines instead partition the verification subject into ordered batches, assign each batch to an independent verifier, and merge the checked findings. `direct_pages` forms page ranges over the original document. `page_text` and `text_blocks` extract page-wise content and text blocks directly from the original document, then batch the resulting pages or blocks. `claims` and `questions` instead further extract atomic items from the page-wise content and batch them by item order. Each verifier handles only its target batch, retrieves additional document context only when needed, and searches the exam data directly. Horizontal decomposition therefore controls the workload of the verification call itself, at the cost of distributing document-wide context across multiple verifiers.

Table 11: Comparison of the direct baseline and the task-decomposition strategies, before visual specialization and review, on the complete 30-page deck. "Verification unit" is the material assigned to one final verifier, and "Evidence" is the exam evidence available to it. "Found" lists the three run counts out of 41; recall is the mean over runs. Bold marks the highest mean recall within each decomposition family.

Strategy	Verification unit	Evidence	Found	Mean recall
Vertical: sequential stage decomposition and direct baseline				
`direct`	Full original document	Full exam data	30, 28, 30	71.5%
`direct_groundtruth`	Full original document	Retrieved information	31, 32, 35	79.7%
`content`	Full page extraction	Full exam data	33, 33, 34	81.3%
`content_groundtruth`	Full page extraction	Retrieved information	34, 32, 34	81.3%
`claims`	Complete claim set	Full exam data	37, 35, 31	**83.7%**
`claims_groundtruth`	Complete claim set	Retrieved information	31, 28, 31	73.2%
`questions`	Complete question set	Full exam data	32, 36, 35	**83.7%**
`questions_groundtruth`	Complete question set	Retrieved information	33, 25, 35	75.6%
`blocks`	Complete block set	Full exam data	32, 31, 30	75.6%
`blocks_groundtruth`	Complete block set	Retrieved information	29, 32, 34	77.2%
Horizontal: parallel batch decomposition				
`direct_pages`	Native-page batch	Full exam data	36, 36, 37	88.6%
`page_text`	Extracted-page batch	Full exam data	37, 34, 38	88.6%
`text_blocks`	Text-block batch	Full exam data	36, 37, 36	88.6%
`claims`	Claim batch	Full exam data	34, 33, 36	83.7%
`questions`	Question batch	Full exam data	36, 37, 39	**91.1%**

Results and strategy selection. Every tested decomposition has higher mean recall than direct GPT-5.5 verification (71.5%). The strongest vertical strategies, `claims` and `questions`, both reach 83.7%. Separating exam-data retrieval does not consistently help: it raises the direct baseline to 79.7%, leaves `content` unchanged at 81.3%, and lowers `claims` and `questions` to 73.2% and 75.6%, respectively. Offloading retrieval alone therefore provides no general recall advantage in this experiment.

Horizontal batching reaches 83.7–91.1% mean recall. Relative to vertical verification with the full exam data, it improves page-text, block, and question verification by 7.3, 13.0, and 7.3 percentage points, respectively; claim verification is unchanged. Distributing the final verification workload is therefore the more effective design on this deck, although workflow and total compute are not held equal across strategies. Horizontal `questions` has the best mean at this stage (91.1%), and `text_blocks` reaches 88.6%, detecting 36–37 errors per run.

We nevertheless carry the block pipeline forward. Blocks, unlike derived questions, are units of the document itself: they can be linked to visual objects, reviewed independently, and used as a common unit of work across all three document types.

H.2.3 Effectiveness of Visual Specialization and Review

Experimental design. We use a cumulative ablation of the horizontal block pipeline. All variants use the complete 30-page deck, the same reference exam data and 41-error reference set, the same block extraction and batching settings where applicable, and the same manual matching protocol. All verifiers and reviewers use Codex with GPT-5.5.

The baseline is the horizontal `text_blocks` variant. The first variant adds *dedicated visual verification* but disables review. Extraction indexes ordinary-text blocks and visual objects separately; text verifiers check claims originating in ordinary text, while visual verifiers inspect the assigned objects in the rendered pages together with their source-block context. Their checked first-pass findings are then merged. The second variant adds the complete *independent review* stage. A separate reviewer receives each checked first-pass result together with the same assigned blocks or visual objects, relevant document context, page images where applicable, and the exam data. It rechecks every reported item and independently searches the assigned material for omissions, directly removing, repairing, or adding items before the reviewed text and visual findings are merged.

Table 12: Cumulative ablation of visual specialization and independent review on the horizontal block pipeline. "Found" lists the three run counts out of 41; recall is the mean over runs. Bold marks the highest mean recall. The baseline is the same `text_blocks` result as in Table 11.

Variant	Visual verification	Review	Found	Mean recall
`text_blocks` baseline	No	No	36, 37, 36	88.6%
+ dedicated visual verification	Yes	No	40, 38, 37	93.5%
+ independent review	Yes	Yes	39, 39, 39	**95.1%**

Results and interpretation. Adding dedicated visual verification without review raises mean recall from 88.6% to 93.5%, a gain of 4.9 percentage points (two additional reference errors per run on average). Adding independent review raises it by a further 1.6 points to 95.1%, with 39 of 41 errors detected in each run. The complete pipeline improves on direct GPT-5.5 verification by 23.6 points and on the block baseline by 6.5 points. We therefore adopt the block pipeline with dedicated visual verification and independent review for the benchmark.

H.2.4 Precision Evaluation on Benchmark Outputs

Manual inspection. We take the ten Biology slide decks from the main experiment and manually check every reported factual error against the deck and the exam data. The inspected set contains 445 findings, of which 6 are confirmed false positives (1.35%), corresponding to 98.65% precision.

Agent inspection. The remaining subjects and tasks are inspected by the LLM classification pipeline of Appendix H.4.1, which marks false positives while assigning patterns; its manual check on the Biology slides is reported there. Table 13 reports the results.

Results. False positives account for 2.06% of reported findings for slides, 4.89% for reports, and 4.03% for student worksheets. Together with the 98.65% precision on the manually checked subset, these low proportions indicate that false positives have little effect on the factual correctness scores.

Table 13: False-positive inspection of benchmark factual-error findings across the five subjects. "Reported items" counts findings labeled `incorrect`; "FP share" is the false-positive count divided by this total. The Biology slides row was checked manually; the other rows by the classification pipeline.

Subject	Reported items	False positives	FP share
Slides			
Biology (manual)	445	6	1.35%
Chemistry	463	14	3.02%
English Language	333	7	2.10%
Mathematics	420	8	1.90%
Physics	427	8	1.87%
Slides total	**2,088**	**43**	**2.06%**
Reports			
Biology	92	4	4.35%
Chemistry	113	3	2.65%
English Language	102	5	4.90%
Mathematics	125	7	5.60%
Physics	120	8	6.67%
Reports total	**552**	**27**	**4.89%**
Student worksheets			
Biology	337	3	0.89%
Chemistry	309	17	5.50%
English Language	314	16	5.10%
Mathematics	274	18	6.57%
Physics	278	7	2.52%
Student worksheets total	**1,512**	**61**	**4.03%**

H.3 Result Details

Appendix A gives factual-correctness scores and pass rates for every task–model pair. Section 5.1 discusses the differences between tasks.

H.4 Failure Patterns

H.4.1 Classification Method

Figure 23 shows the subsequent inspection and classification pipeline. It collects the final `incorrect` findings while preserving their source records and repeated occurrences. Visual findings are first assessed in batches of at most 25: the inspector reads the linked page images and identifies whether the error is conveyed through text, non-text properties, both, or an unclear mode. The reviewed assessments are merged with the original findings, which are then batched in groups of at most 50. Each classification batch receives the reported claims, errors and corrections, category definitions, and the exam data. The classifier jointly identifies false positives and assigns valid errors to a failure pattern; quantitative errors additionally receive one data-type label and one cause label. Both stages include a deterministic output check, independent review, and a second check, with invalid outputs returned for rework. Final aggregation checks exact item coverage and reports pattern counts, quantitative-label counts, false positives, and unmatched findings. Mistaken findings are marked as false positives; valid findings without a supported category remain unmatched.

Manual check of the pipeline. Pattern assignment is an evidence judgment like factual verification, with more categories, so we checked whether the larger label set reduced judgment quality. For the Biology slides, whose findings were also checked manually for precision (Appendix H.2.4), we manually reviewed the pipeline's validity decisions and pattern assignments. With batches of at most 50 findings and visual batches of at most 25, the review found few clear mistakes. Most disputed validity judgments concerned low-severity issues that could be treated either as false positives under a stricter threshold or as suggestions for improving the document; the few pattern disagreements involved ambiguous boundaries between categories. We therefore use the pipeline for the remaining subjects and tasks.

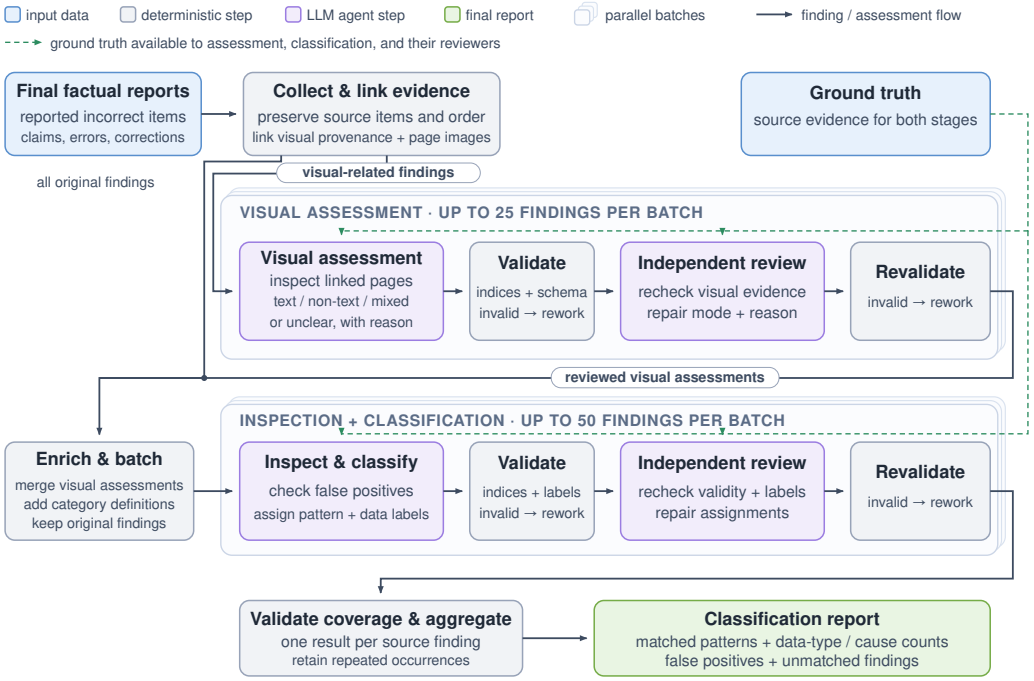


Figure 23: Factual-error inspection and classification pipeline. Reviewed visual assessments inform joint false-positive inspection and pattern assignment; only checked, reviewed results enter the aggregate report.

H.4.2 Distribution

Overall distribution. Across the five subjects and ten agents, 4,015 of 4,152 reported findings are assigned to a pattern; 131 false positives and six unmatched findings (all six in student worksheets) are excluded. Within each task, percentages use the classified findings of the corresponding group as their denominator (Table 14).

Table 14: Factual-error families by task. Counts are classified findings; percentages show each family's share of the classified findings for that task or for all tasks combined.

Family	Slides	Reports	Worksheets	All
1. Source misrepresentation	247 (12.1%)	135 (25.7%)	383 (26.5%)	765 (19.1%)
2. Association / reference misuse	143 (7.0%)	77 (14.7%)	212 (14.7%)	432 (10.8%)
3. Quantitative errors	1,306 (63.9%)	212 (40.4%)	140 (9.7%)	1,658 (41.3%)
4. Interpretation / reasoning	108 (5.3%)	87 (16.6%)	148 (10.2%)	343 (8.5%)
5. Chart / diagram representation	58 (2.8%)	0 (0.0%)	0 (0.0%)	58 (1.4%)
6. Missing / incomplete content	129 (6.3%)	5 (1.0%)	392 (27.1%)	526 (13.1%)
7. Internal contradictions	54 (2.6%)	9 (1.7%)	170 (11.8%)	233 (5.8%)
Classified findings	2,045 (100.0%)	525 (100.0%)	1,445 (100.0%)	4,015 (100.0%)

By task. Quantitative errors lead in slides and reports, whereas missing content (27.1%) and source misrepresentation (26.5%) together account for 53.6% of worksheet findings. Reports have larger shares of source misrepresentation (25.7%) and interpretation and reasoning errors (16.6%) than slides (12.1% and 5.3%), and internal contradictions are mostly a worksheet problem (11.8%, against 2.6% in slides and 1.7% in reports). Section 5.2 analyzes the causes of the quantitative errors.

By subject. Table 15 breaks the families down by subject. Quantitative errors are the largest family for every subject in slides (56.8–68.3% of each subject's findings) and in reports (33.6–48.2%). The secondary families differ by subject. Association and reference misuse is far more common in

English than elsewhere: 12.9% of English slide findings against 3.9% for Biology, 25.8% of English report findings against 9.3–15.9% for the other subjects, and 32.6% of English worksheet findings against 2.0% for Mathematics. Interpretation and reasoning errors concentrate in Physics slides (10.5%) and in Chemistry and Mathematics reports (21.8% and 21.2%). Missing content is stable across worksheet subjects at 22.5–32.3%.

Table 15: Classified factual findings by task and subject, pooled over ten agents. Columns 1–7 denote the families defined below. † marks the manually checked Biology slides; the other rows use the classification pipeline.

Subject	1	2	3	4	5	6	7	Total
Slides								
Biology†	56	17	300	6	15	37	8	439
Chemistry	47	33	282	33	18	19	17	449
English	35	42	213	4	8	15	9	326
Mathematics	64	30	234	21	9	43	11	412
Physics	45	21	277	44	8	15	9	419
Task total	**247**	**143**	**1,306**	**108**	**58**	**129**	**54**	**2,045**
Reports								
Biology	19	14	40	13	0	0	2	88
Chemistry	34	14	37	24	0	0	1	110
English	23	25	41	5	0	1	2	97
Mathematics	37	11	40	25	0	2	3	118
Physics	22	13	54	20	0	2	1	112
Task total	**135**	**77**	**212**	**87**	**0**	**5**	**9**	**525**
Worksheets								
Biology	135	30	17	38	0	75	39	334
Chemistry	74	47	37	31	0	81	22	292
English	48	97	10	18	0	88	37	298
Mathematics	70	5	49	29	0	62	40	255
Physics	56	33	27	32	0	86	32	266
Task total	**383**	**212**	**140**	**148**	**0**	**392**	**170**	**1,445**
All tasks	**765**	**432**	**1,658**	**343**	**58**	**526**	**233**	**4,015**

By agent. Table 16 breaks the same findings down by the agent that produced the document. Totals range from 86 classified findings for GPT-5.5 to 773 for MiniMax M3. Quantitative errors are the largest slide family for all ten agents (39.0% for GPT-5.5 to 76.5% for MiMo V2.5 Pro) and the largest or joint-largest report family for eight. Secondary concentrations are agent-specific: source misrepresentation is 25.2% of Claude Opus 5's slide findings against 4.9–18.6% for the others, and association and reference misuse is 21.2% of Gemini 3.5 Flash's against 1.8–9.0%. Worksheet profiles diverge most: source misrepresentation leads for four agents (52.2% for Gemini 3.5 Flash), missing content for another four (43.9% for MiMo V2.5 Pro and 41.5% for DeepSeek V4 Pro), internal contradictions for Claude Opus 5, and association and reference misuse for Qwen 3.7 Max.

Table 16: Classified factual findings by task and producing agent, pooled over five subjects. Columns 1–7 denote the families defined below. Totals match Table 15.

Agent (harness / model)	1	2	3	4	5	6	7	Total
Slides								
Codex / GPT-5.5	2	1	16	3	9	10	0	41
OpenCode / Grok 4.5	2	1	23	5	4	3	2	40
Claude Code / Claude Opus 5	41	3	97	14	4	1	3	163
OpenCode / Gemini 3.5 Flash	31	39	83	9	1	17	4	184
OpenCode / GLM 5.2	44	17	129	9	13	19	6	237
OpenCode / MiMo V2.5 Pro	20	12	166	4	7	6	2	217
OpenCode / Kimi K2.6	26	22	169	14	2	6	6	245
OpenCode / DeepSeek V4 Pro	18	8	167	2	2	28	2	227
OpenCode / Qwen 3.7 Max	32	15	260	25	4	7	10	353
OpenCode / MiniMax M3	31	25	196	23	12	32	19	338
Task total	**247**	**143**	**1,306**	**108**	**58**	**129**	**54**	**2,045**
Reports								
Codex / GPT-5.5	0	3	0	0	0	0	1	4
OpenCode / Grok 4.5	5	8	7	5	0	0	2	27
Claude Code / Claude Opus 5	11	9	15	7	0	0	2	44
OpenCode / Gemini 3.5 Flash	9	3	24	5	0	2	0	43
OpenCode / GLM 5.2	21	4	27	10	0	1	0	63
OpenCode / MiMo V2.5 Pro	18	4	31	7	0	0	0	60
OpenCode / Kimi K2.6	10	4	19	9	0	0	0	42
OpenCode / DeepSeek V4 Pro	6	15	15	11	0	0	1	48
OpenCode / Qwen 3.7 Max	24	11	33	8	0	1	0	77
OpenCode / MiniMax M3	31	16	41	25	0	1	3	117
Task total	**135**	**77**	**212**	**87**	**0**	**5**	**9**	**525**
Worksheets								
Codex / GPT-5.5	15	2	5	5	0	14	0	41
OpenCode / Grok 4.5	38	5	16	14	0	25	9	107
Claude Code / Claude Opus 5	14	4	21	11	0	14	22	86
OpenCode / Gemini 3.5 Flash	35	8	6	14	0	3	1	67
OpenCode / GLM 5.2	46	15	10	14	0	60	16	161
OpenCode / MiMo V2.5 Pro	19	29	4	11	0	50	1	114
OpenCode / Kimi K2.6	22	20	11	21	0	38	4	116
OpenCode / DeepSeek V4 Pro	36	27	8	13	0	88	40	212
OpenCode / Qwen 3.7 Max	35	69	5	6	0	51	57	223
OpenCode / MiniMax M3	123	33	54	39	0	49	20	318
Task total	**383**	**212**	**140**	**148**	**0**	**392**	**170**	**1,445**
All tasks	**765**	**432**	**1,658**	**343**	**58**	**526**	**233**	**4,015**

By pattern. Table 17 gives counts for the individual patterns defined below; quantitative errors additionally receive independent data-type and cause labels (Table 18). Two families are dominated by one pattern in every task: object–label–content mismatches (2.1) are 83.1–89.5% of Family 2, and self-description–content mismatches (7.1) are 88.9–100% of Family 7. Family 6 splits by task: missing information or steps (6.4) is 47.2% of its worksheet findings but 16.3% of its slide findings, where omitted list items (6.2, 40.3%) and truncated content (6.1, 38.8%) lead instead.

Table 17: Detailed factual-error patterns. Each classified finding contributes to exactly one row. Family 3 receives additional data-type and cause labels in Table 18.

Pattern	Slides	Reports	Worksheets	All
1.1 Incorrect restatement of facts / values	151	69	163	383
1.2 Alteration of original meaning	45	18	53	116
1.3 False source presence / absence claims	51	48	167	266
2.1 Object–label–content mismatch	128	64	183	375
2.2 Misuse of reference materials	15	13	29	57
3 Data processing / quantitative expression	1,306	212	140	1,658
4.1 Incorrect concepts / relationships	50	20	68	138
4.2 Invalid premises / reasoning steps	10	25	62	97
4.3 Overstated generalizations / conclusions	48	42	18	108
5.1 Incorrect value-to-graphic mapping	38	0	0	38
5.2 Incorrect legends / visual encodings	9	0	0	9
5.3 Incorrect annotation placement / targeting	10	0	0	10
5.4 Incorrect diagram content / structure	1	0	0	1
6.1 Truncated content	50	0	111	161
6.2 Omitted list items / fields	52	4	84	140
6.3 Incomplete cases / alternatives	6	0	12	18
6.4 Missing necessary information / steps	21	1	185	207
7.1 Self-description–content mismatch	54	8	161	223
7.2 Conflicting statements	0	1	9	10
Total	2,045	525	1,445	4,015

Implications for verification. These distributions call for checks beyond recomputing numbers. A calculation must be traced to the metric and population the claim states, since a mismatched scope is the most common identifiable cause of quantitative errors. Restated facts must be traced back to the source with their values and qualifications intact, and valid information must stay attached to the right student, question, or object. Coverage a document promises, such as a listed set of items or a stated number of exercises, must be compared with what it delivers, and statements in different parts of a worksheet must be read together, since most contradictions are between a self-description and the content. Conclusions must stay within what the records support, and charts must be read for the values and relationships they encode, not only for legibility.

H.4.3 Pattern Catalog

Each pattern below has one or two examples, except that Family 3 illustrates its data types instead. The examples show the classification boundaries and identify the agent, task, subject, and page. Student identifiers are local to each subject's class. Quoted fragments preserve the reported wording. Slides are shown in full; report and student-worksheet images show the relevant text blocks. Red boxes mark the statements or visual elements at issue.

H.4.4 Family 1: Misrepresentation of source information

These errors arise when a generated document restates supplied information incorrectly. The comparison is with the source answer, rubric, or recorded fact, rather than with a newly derived quantity.

1.1 Incorrect restatement of facts or values. A quotation or restatement changes a source fact, value, attribute, or symbol. A directly available numerical value belongs here when its meaning and scope are unchanged; recalculation or a change of metric belongs to Family 3.

Example (Slides; Biology; Claude Opus 5; p. 28; Figure 24). The deck describes Student 010 as having full marks on Q5(b)(ii), although the recorded score is $2/3$. The error directly changes an available source result.

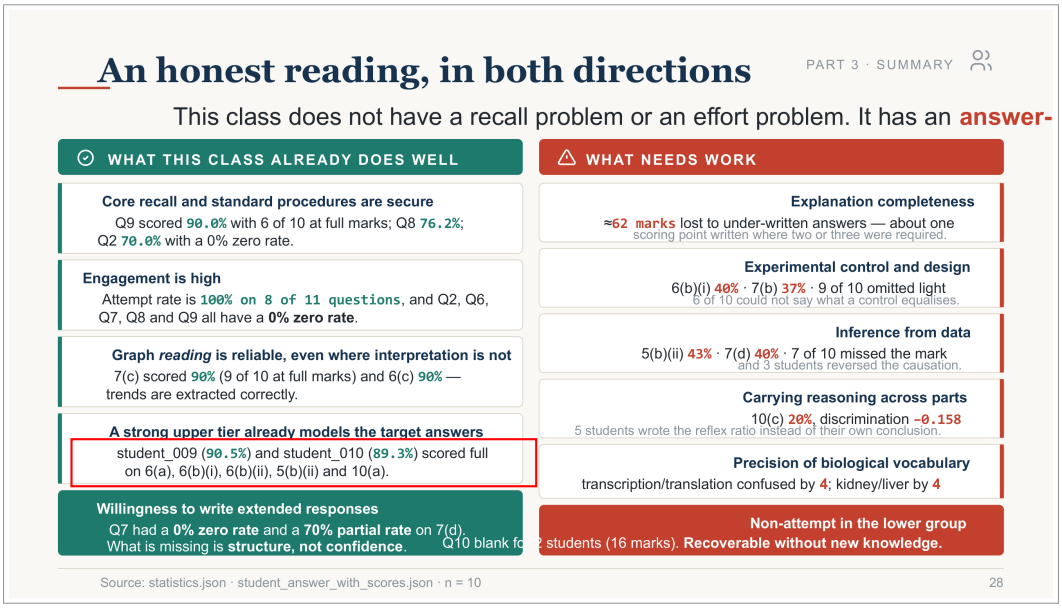


Figure 24: Pattern 1.1: Claude Opus 5, Biology slides, p. 28. The slide reports full marks for Student 010 on Q5(b)(ii), whereas the source score is $2/3$.

1.2 Alteration of original meaning. A summary, interpretation, or paraphrase changes the source's meaning, requirements, conditions, or qualifications, even if it retains some of the original vocabulary.

Example (Slides; Biology; Gemini 3.5 Flash; p. 18; Figure 25). Q7(b) is described as an apparatus task to measure respiration rate. The rubric instead asks for an apparatus to show whether the sample can respire. The paraphrase changes detection into quantitative measurement.

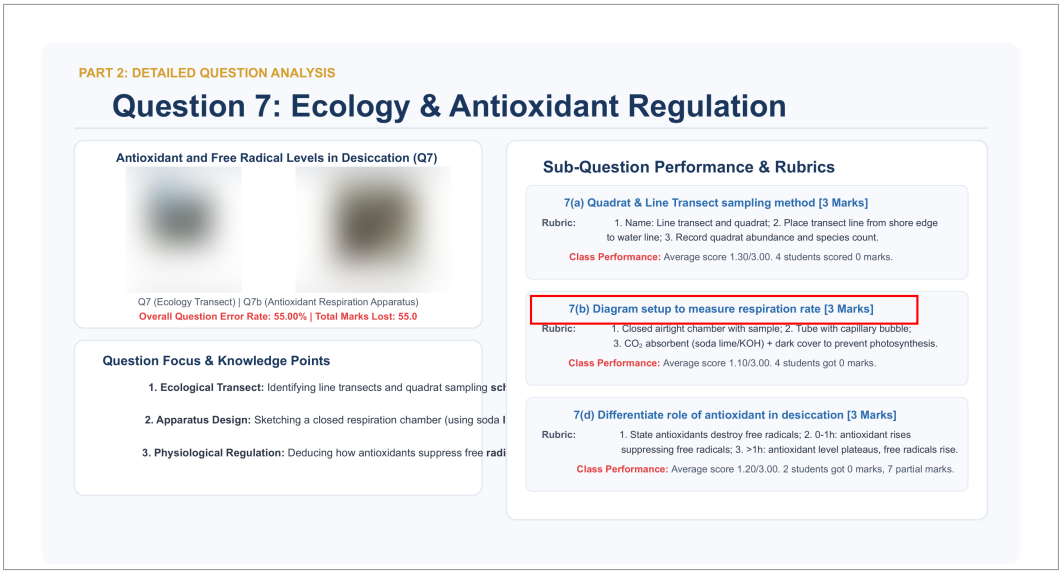


Figure 25: Pattern 1.2: Gemini 3.5 Flash, Biology slides, p. 18. The slide changes a task about detecting respiration into one about measuring its rate. The two source-page crops are blurred.

1.3 Incorrect claims about source content being present or absent. The generated document claims that source material exists when it does not, or denies material that is present. This includes treating an incorrect answer as a blank answer and describing features of an unattempted answer.

Example (Report; English Language; GLM 5.2; p. 3; Figure 26). Student 004 is described as having left Q9 blank, together with Q11, Q12, and Q31. The source records "D" for Q9; only the other three are empty.

Floor band (scores 19–30; p25≈27): 001(19), 004(23), 003(26), 002(30). Linked: 0 on Q10/Q38/Q7; ≤1 on Q32; blanks (004 empty on Q9/Q11/Q12/Q31; 002 "Because" only on Q10). Fail P1+P2+P3. Next step: small-group; start P1 two-box on a fresh short contrastive text at sentence level, paired with the P2 synonym-test on a known-vocabulary paragraph; threshold = both propositions on one fresh item before advancing; run the P2 cue-vocabulary check to settle the rival for these four.

Figure 26: Pattern 1.3: GLM 5.2, English Language report, p. 3. The report calls Student 004's Q9 answer blank, although the recorded answer is "D".

Example (Slides; Biology; GLM 5.2; p. 17; Figure 27). The slides claim that Q5(b) has "no itemised scoring points in the rubric" and therefore report only aggregate marks lost. The rubric itemizes Q5(b)(i), (ii), and (iii), worth 2, 3, and 1 marks, with scoring rationales. The error denies source content that is present.

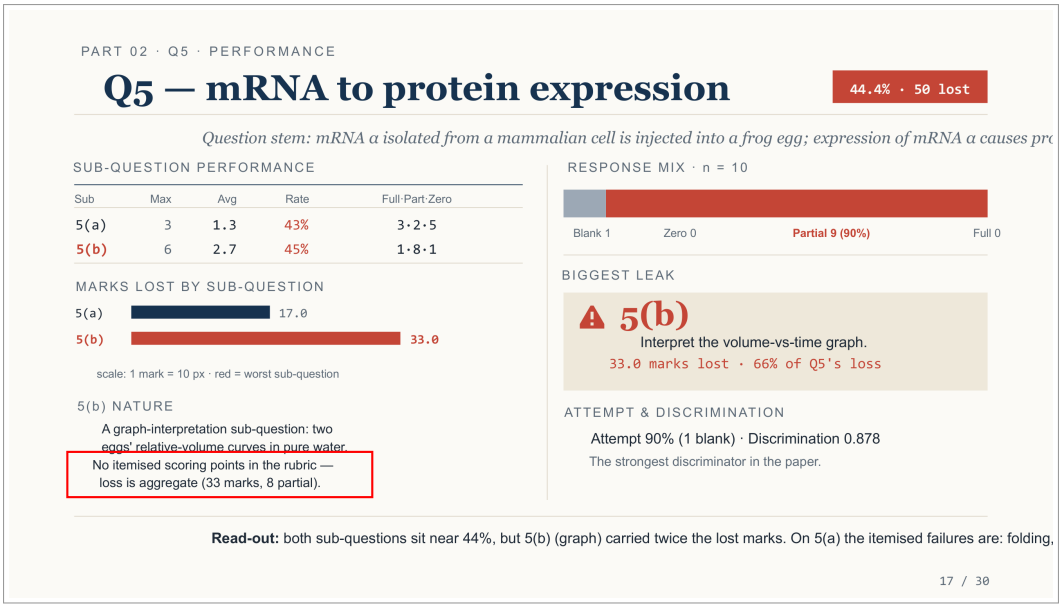


Figure 27: Pattern 1.3: GLM 5.2, Biology slides, p. 17. The slide states that the rubric has no itemized scoring points for Q5(b).

H.4.5 Family 2: Information Association and Reference Material Errors

These errors attach information to the wrong object or use material with an incorrect source identity. A genuine value or quotation can therefore still be wrong in its assigned context.

2.1 Mismatch between objects, labels, and content. A name, identifier, category, location, or description is associated with the wrong object or content. The central error is the association, rather than the intrinsic validity of the value or statement.

Example (Slides; Mathematics; Gemini 3.5 Flash; p. 4; Figure 28). Student 010 is assigned $90/105$. That score belongs to Student 009; Student 010 scored $89/105$.

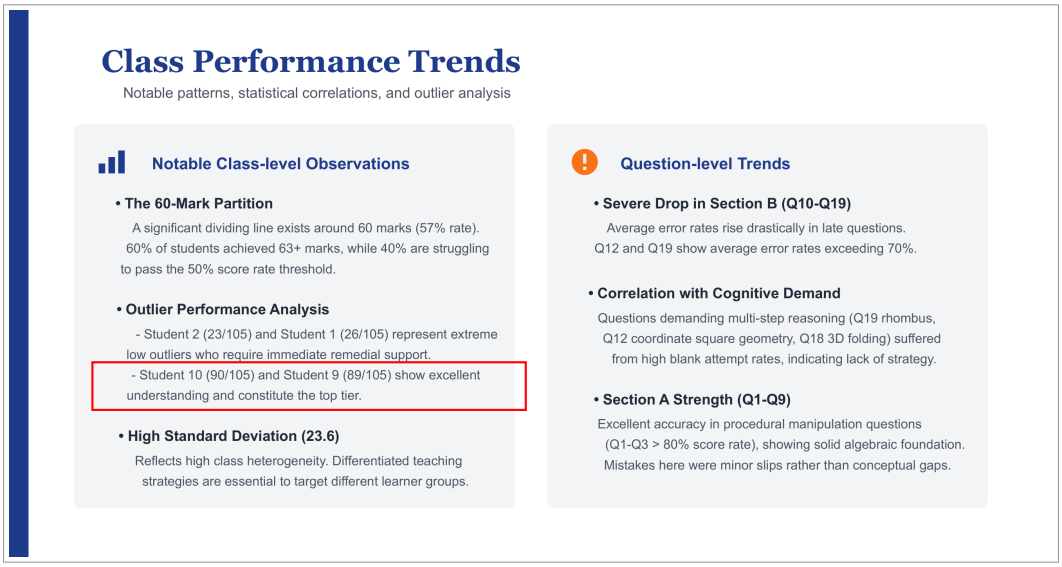


Figure 28: Pattern 2.1: Gemini 3.5 Flash, Mathematics slides, p. 4. The slide assigns Student 009's score of $90/105$ to Student 010, who scored $89/105$.

Example (Slides; Biology; Gemini 3.5 Flash; p. 6; Figure 29). The question-performance table labels Q8 as "Plant hormones." Q8 concerns disease burden, hepatitis B vaccination, and liver cancer; plant hormones belong to Q9. A valid topic label is attached to the wrong question.



Figure 29: Pattern 2.1: Gemini 3.5 Flash, Biology slides, p. 6. The Q8 row is labeled "Plant hormones," the topic of Q9.

2.2 Misuse of reference materials. An excerpt, image, or other reference is used as evidence for the wrong object, answer, source, or purpose. Unlike a general label mismatch, the error concerns the provenance or use of the referenced material itself.

H.4.6 Family 3: Data processing and quantitative expression errors

This family covers errors in deriving, attributing, or interpreting quantitative information. Each finding receives one data-type label (D1–D6) and one cause label (C1–C5). These dimensions answer

different questions: what result is wrong, and what supported fault explains it. They do not add findings to the total. A wrong count alone, for example, does not establish that the agent performed an incorrect set intersection.

Table 18: The two independent label dimensions of quantitative errors. Each dimension sums to the same 1,658 findings; its counts must not be added to the other dimension or to the primary pattern counts.

Label	Slides	Reports	Worksheets	All
Data type				
D1 Counts / frequencies	544	68	10	622
D2 Joint / distinct counts	178	3	0	181
D3 Amounts / numerical summaries	233	19	40	292
D4 Ratios / rates / shares	139	26	8	173
D5 Orderings / selections / characterizations	212	96	82	390
D6 Other / unclear data types	0	0	0	0
Total for this dimension	1,306	212	140	1,658
Error cause				
C1 Metric definition / scope	417	44	12	473
C2 Record handling / membership	52	13	13	78
C3 Numerical conditions / comparisons	183	78	72	333
C4 Precision / approximation	3	3	3	9
C5 Other / unclear causes	651	74	40	765
Total for this dimension	1,306	212	140	1,658

Data types and representative examples. Each data type has one example; D6 is a fallback with no observed instances.

D1: Counts and frequencies.

An incorrect number of entities, records, or occurrences in a population or category. Ordinary conditional tallies remain D1; explicitly joint or distinct membership counts use D2.

Example (Slides; Biology; Claude Opus 5; p. 22; Figure 30). The slides state that six of nine students missed Q7(d)'s post-1 h plateau evidence point, losing six marks. The source records seven misses among ten students and seven marks lost. Student 001's equivalent rationale uses different wording; omitting this record understates the count, giving cause C2.

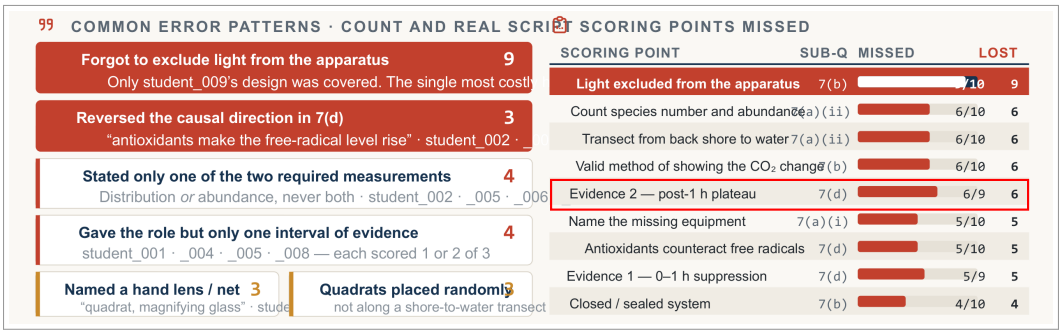


Figure 30: Data type D1: Claude Opus 5, Biology slides, p. 22. The slide undercounts students who missed Q7(d) by omitting an equivalent answer.

D2: Joint and distinct counts.

An incorrect number of shared or distinct entities across groups, conditions, or events, such as an intersection or union count.

Example (Slides; Physics; Gemini 3.5 Flash; p. 22; Figure 31). Eight students are described as mentioning different harmonic strengths but omitting how harmonics superimpose to form waveforms. Eight missed the superposition point, but only five also earned the harmonic-strength point; Students 001, 004, and 006 missed both. Substituting the broader population for the specified intersection gives cause C1.

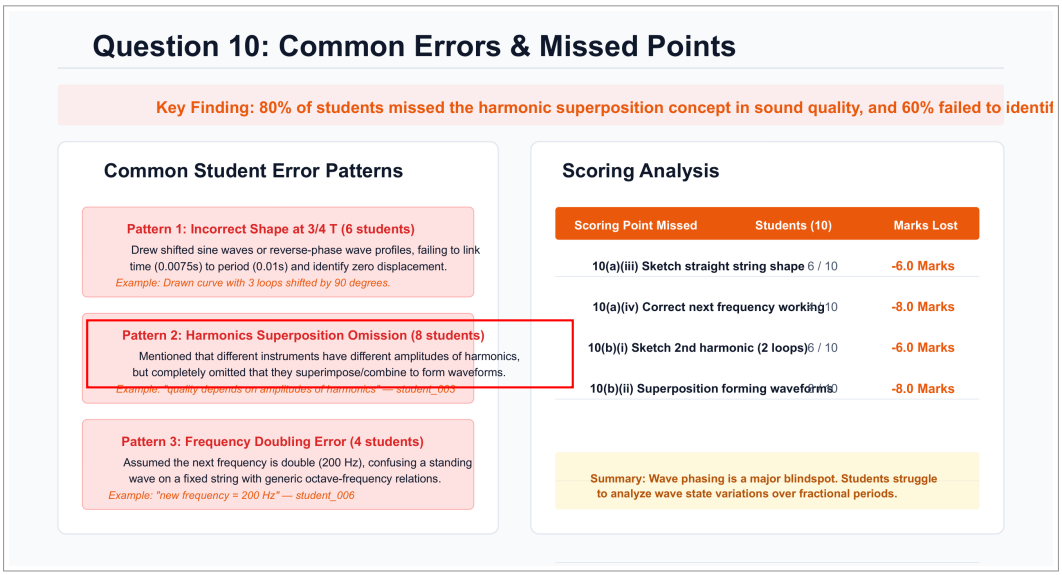


Figure 31: Data type D2: Gemini 3.5 Flash, Physics slides, p. 22. The slide counts eight students for a condition that only five satisfy.

D3: Amounts and numerical summaries.
An incorrect derived amount, total, average, difference, or converted value. This concerns the numerical result rather than a relative share.

Example (Student worksheet; Mathematics; Grok 4.5; p. 4; Figure 32). A practice answer gives $BC^2 = 81 + 49 - 126\cos 70°$ and then $BC \approx 9.4$ to one decimal place. The expression gives $BC \approx 9.3223$, which rounds to 9.3. The incorrect rounding gives cause C4.

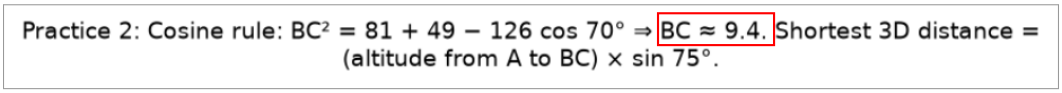


Figure 32: Data type D3: Grok 4.5, Mathematics student worksheet, p. 4. Answer excerpt rotated for readability.

D4: Ratios, rates, and shares.
An incorrect relative quantity, such as a percentage, ratio, rate, relative change, or distribution share. A tally expressed as a number out of a population remains a count when the erroneous result is the number of entities.

Example (Report; Biology; MiniMax M3; p. 1; Figure 33). Q4(a) is described as "35% zero." The 35% is the average score rate, $0.7/2$; five of ten students scored zero, giving 50% zero. Substituting the measured property makes this a C1 error.

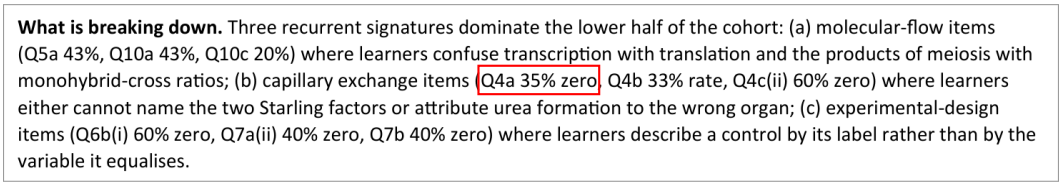


Figure 33: Data type D4: MiniMax M3, Biology report, p. 1. The report mistakes a 35% average score rate for the proportion scoring zero.

D5: Orderings, selections, and quantitative characterizations.
An incorrect ranking, selected set, grouping, or verbal description of quantitative relationships. If only a numerical value within such a statement is wrong, its type is D1–D4 instead.

Example (Student worksheet; Biology; Claude Opus 5; pp. 19–20; Figure 34). Three focus questions are described as class strengths on which Student 010 underperformed. On Q4(b), the student's $1/3$ ties the class average; on Q5(a), $2/3$ exceeds the class's $1.3/3$. Only Q7(d) has the stated direction of difference. Misinterpreting these comparisons and the tie gives cause C3.

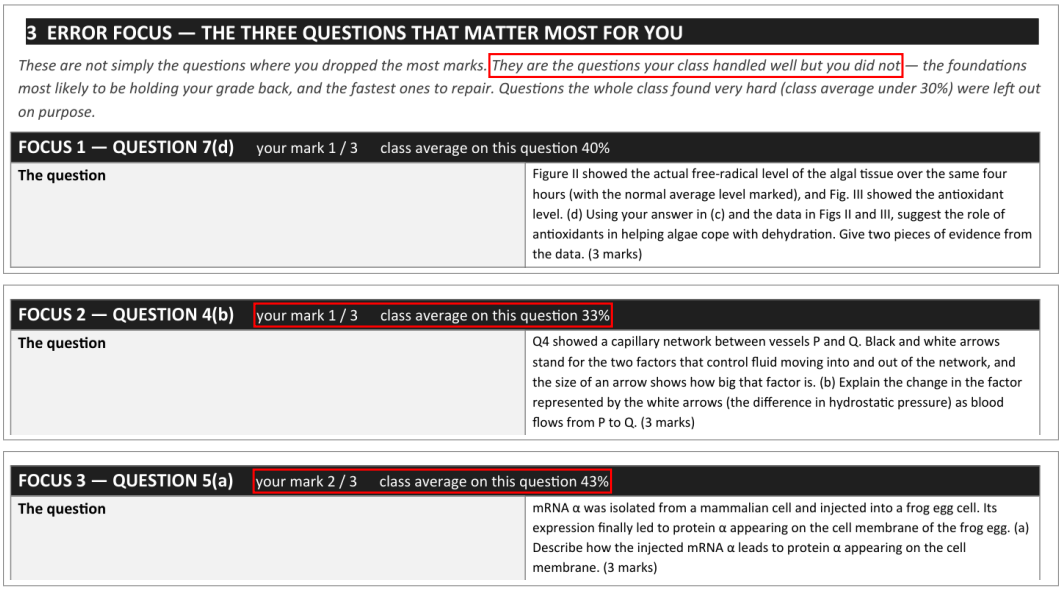


Figure 34: Data type D5: Claude Opus 5, Biology student worksheet, pp. 19–20. The selection claim and all three focus questions are shown with the student's marks and class averages. The marked comparisons for Q4(b) and Q5(a) contradict the stated underperformance.

D6: Other or unclear data types.
The erroneous quantitative result has another or unidentified form. No finding in the main experiment receives this label, so no example is supplied.

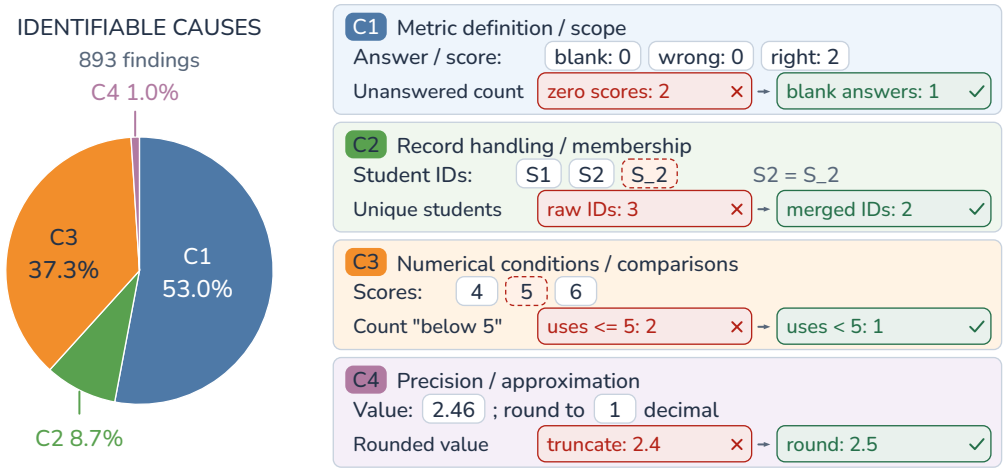


Figure 35: Identifiable causes of quantitative factual errors. The pie chart shows shares of the 893 findings with a specific cause label; the four schematic examples illustrate the categories.

Error causes and their boundaries. The cause is the best-supported fault in deriving, attributing, or interpreting the selected result, inferred from the document and the source evidence. The examples above illustrate how these causes are assigned independently of data type.

C1: Metric definition or scope errors.
The wrong property, population, inclusion criterion, aggregation level, unit, or numerator/denominator meaning is used. The D2 example substitutes a broader population for the required intersection; the D4 example substitutes mean attainment for the proportion of zero-score students.

C2: Record handling or membership combination errors.
Records, fields, labels, or memberships are matched or combined incorrectly within the intended metric and scope, introducing missing, repeated, or wrongly associated contributions. This includes faulty unions, intersections, and deduplication. The D1 example omits a semantically equivalent rationale record with different wording.

C3: Numerical condition or comparison errors.
A numerical relationship is applied or interpreted incorrectly, including inequality endpoints, bounds, ordering, and ties. The D5 example treats a tie and a student advantage as student deficits.

C4: Precision or approximation errors.
Rounding, numerical truncation, or approximation incorrectly changes a value, or the stated precision misrepresents its accuracy. The D3 example reports the wrong one-decimal approximation despite a correct formula.

C5: Other or unclear causes.
The supported cause is outside C1–C4 or cannot be identified from the evidence.

H.4.7 Family 4: Content Interpretation and Reasoning Errors

This family covers incorrect domain explanations, unsupported inferential steps, and conclusions whose strength exceeds the evidence. Direct source miscopying and quantitative derivation are handled by Families 1 and 3.

4.1 Incorrect explanations of concepts or factual relationships. A stated concept, principle, mechanism, or relationship is false according to the supplied evidence or the relevant domain knowledge.

Example (Slides; Mathematics; MiniMax M3; p. 7; Figure 36). The slide claims that "losing 1 mark on Q19 hurts more than 1 mark on Q3." Both losses reduce the paper total by one mark. Q19's larger maximum score and greater overall loss do not make each individual mark more valuable.

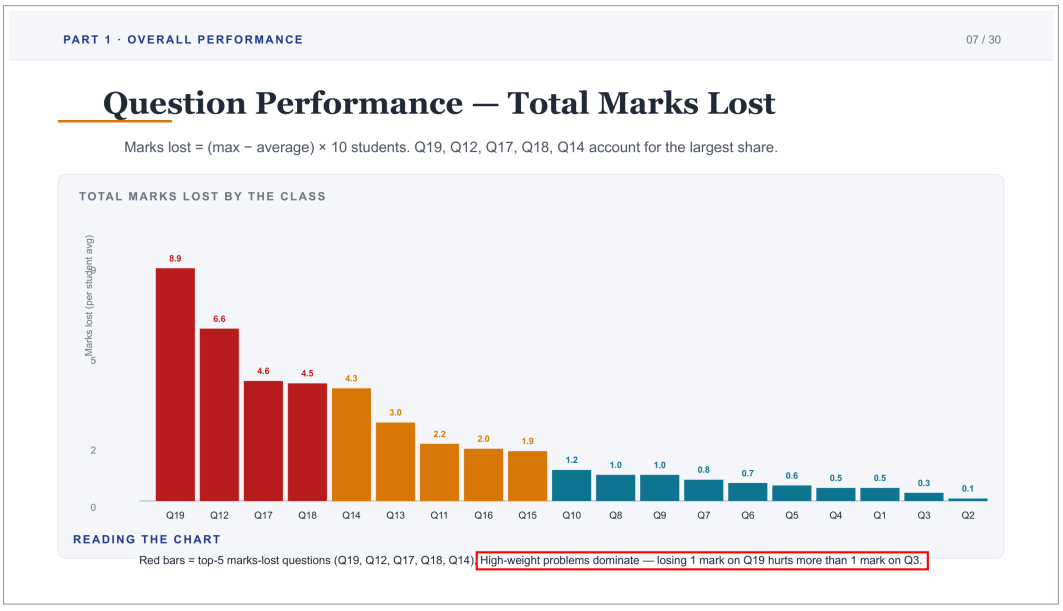


Figure 36: Pattern 4.1: MiniMax M3, Mathematics slides, p. 7. The slide claims one lost mark on Q19 has more effect on the total than one lost mark on Q3.

4.2 Invalid premises or reasoning steps. A conclusion or task depends on a false or unestablished premise, an invalid logical step, or conditions that cannot jointly hold. The error concerns the support for the reasoning, beyond whether its conclusion sounds plausible.

Example (Student worksheet; Mathematics; Grok 4.5; p. 11; Figure 37). A practice task specifies $H(x) = (3x^2 + x + p)(x^2 + rx - 8)$, constant term 24, $H(-1) = -18$, and factor $x - 2$. The constant term requires $p = -3$; the remaining conditions then require both $r = -25$ and $r = 2$. The assumed mathematical object cannot satisfy the stated premises.

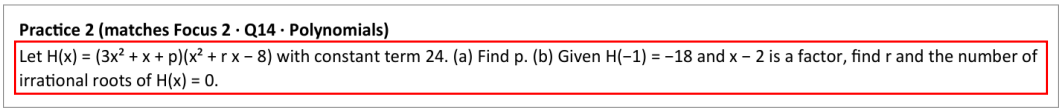


Figure 37: Pattern 4.2: Grok 4.5, Mathematics student worksheet, p. 11. The task gives mutually incompatible conditions for the polynomial $H(x)$.

4.3 Overstated generalizations, attributions, or conclusions. A statement presents a relationship, explanation, or method as universal, necessary, exhaustive, or exclusive when the evidence supports a weaker claim or permits alternatives.

Example (Slides; Chemistry; Claude Opus 5; p. 14; Figure 38). The slides claim, "Every Q8 loss traces to a recording habit." The source also contains incorrect chemistry and calculations: Student 004 reverses the bond-energy comparison, and Student 007 identifies an incorrect formed bond. These counterexamples do not support attributing every lost mark to recording habits.

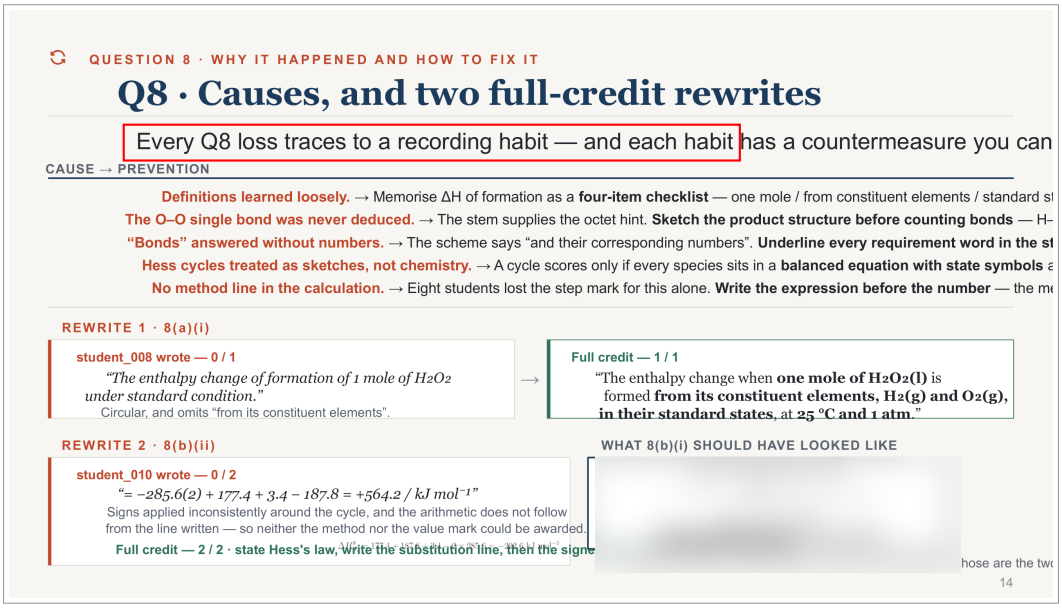


Figure 38: Pattern 4.3: Claude Opus 5, Chemistry slides, p. 14. The slide attributes every Q8 loss to recording habits despite other documented causes. The source marking-scheme crop is blurred.

H.4.8 Family 5: Chart and diagram representation errors

These patterns concern factual meaning carried by graphical position, encoding, annotation, or structure. They exclude purely aesthetic defects. An incorrect underlying number is assigned to Family 1 or 3; a reference image from the wrong source belongs to 2.2. All 58 findings in this family are from slides; the report and student-worksheet runs contain none.

5.1 Incorrect mapping of values to graphics. Under the stated graphical coordinate system, a point, bar, size, proportion, or curve misrepresents its corresponding value. Correct text elsewhere does not repair an incorrectly drawn mark.

Example (Slides; Biology; GPT-5.5; p. 4). In the sorted-score plot, S3 is positioned at approximately 25 marks. The third sorted source score is 31. The point therefore misrepresents the recorded value (Figure 39).

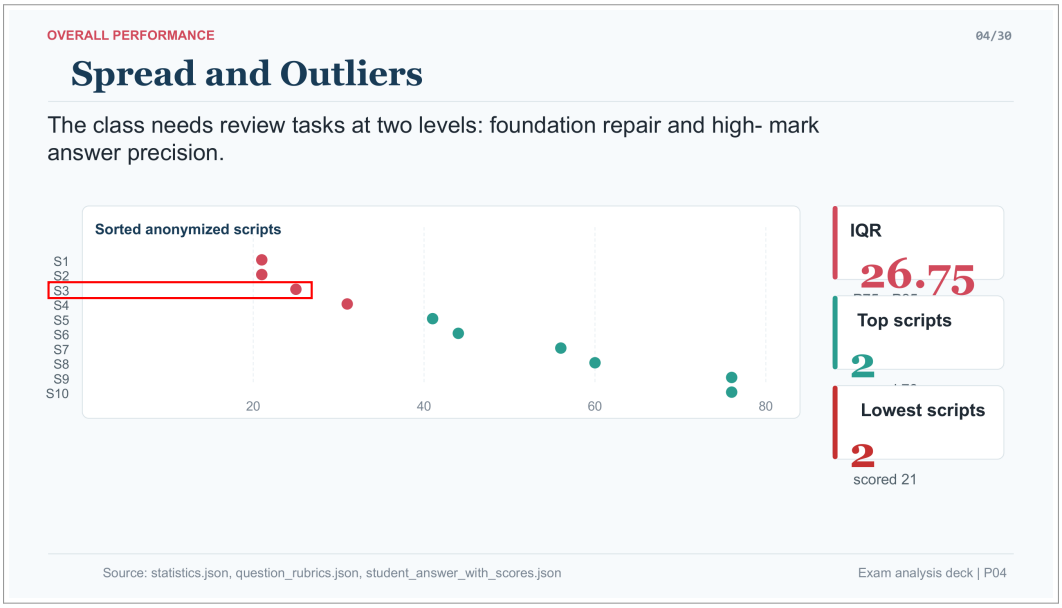


Figure 39: Pattern 5.1: GPT-5.5, Biology slides, p. 4. S3 is drawn near 25 marks, although the third sorted source score is 31.

5.2 Incorrect legends or visual encodings. A color, symbol, line style, fill, or legend assigns an incorrect factual meaning relative to the source values, stated categories, or relevant conventions. The quantity may be correctly printed while its encoding is wrong.

Example (Slides; Mathematics; MiMo V2.5 Pro; p. 5). Q14's average score rate is correctly printed as 46.3%, but the bar is orange, which the legend defines as 50–80%. It should be red under the "Below 50%" category (Figure 40).

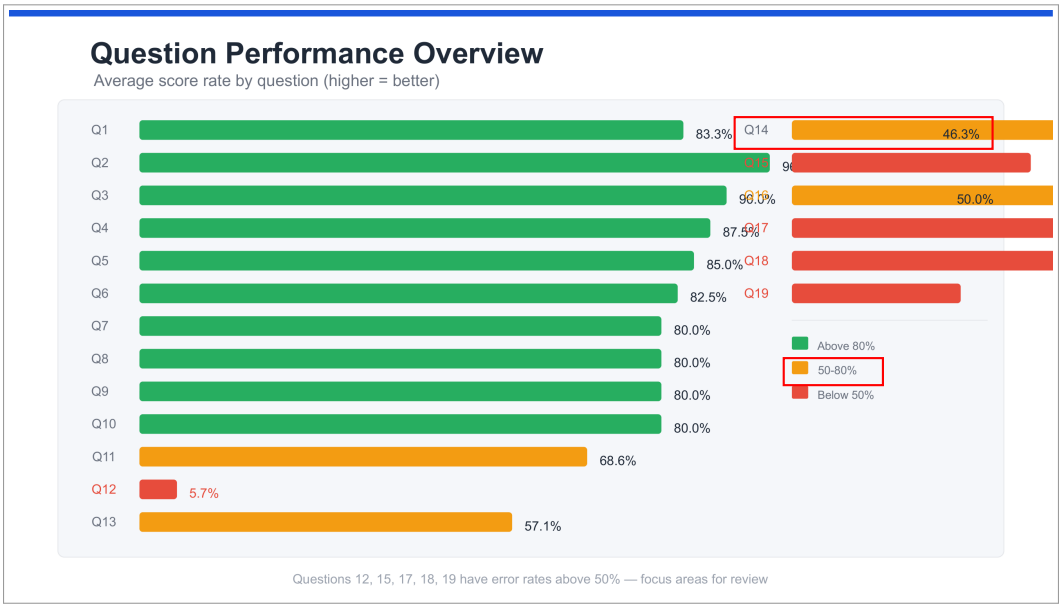


Figure 40: Pattern 5.2: MiMo V2.5 Pro, Mathematics slides, p. 5. Q14 is labeled 46.3% but colored orange, whose legend means 50–80%.

5.3 Incorrect annotation placement or targeting. An annotation, reference line, highlight, or outlined region points to the wrong location, covers the wrong interval, or emphasizes the wrong object. The error lies in what the annotation spatially denotes.

Example (Slides; Biology; MiniMax M3; p. 4). The label "Median 43.5" agrees with the source statistic, but its dashed guide is positioned over the 30–39 band. The median belongs in the 40–49 band (Figure 41).

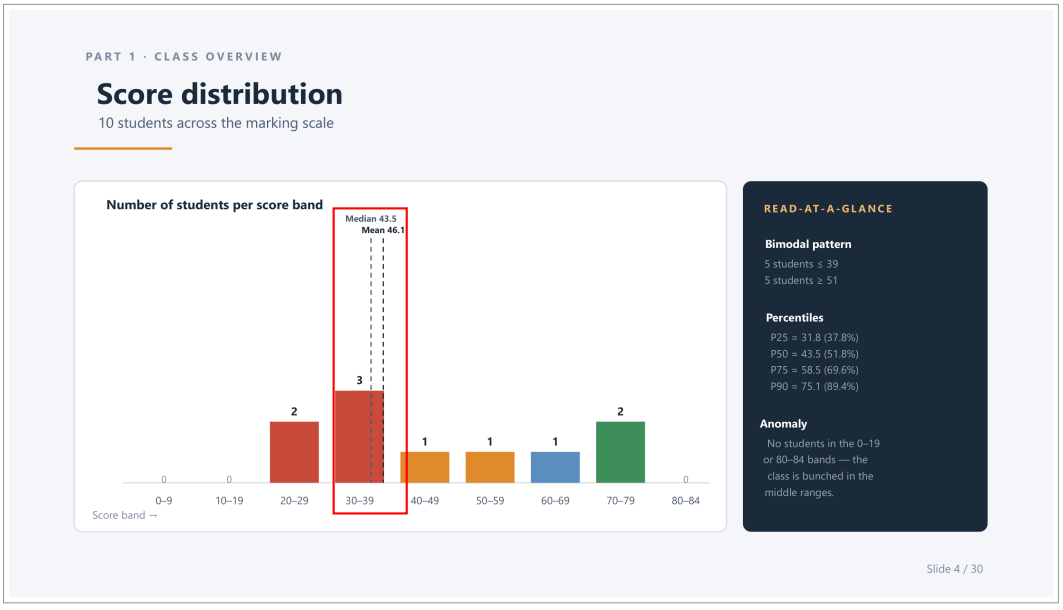


Figure 41: Pattern 5.3: MiniMax M3, Biology slides, p. 4. The dashed median guide is drawn over 30–39 despite its correct label of 43.5.

5.4 Incorrect diagram content or structure. A diagram depicts incorrect components, forms, spatial relations, connections, directions, or processes. The represented structure itself is factually wrong, rather than merely hard to read.

Example (Slides; Chemistry; MiniMax M3; p. 14). The Compound A drawing is labeled propanoic acid and has the correct nearby formula CH_3CH_2COOH, yet it connects CH_3 directly to the carboxyl carbon. The required intervening CH_2 group is missing, so the depicted connectivity does not represent propanoic acid (Figure 42).

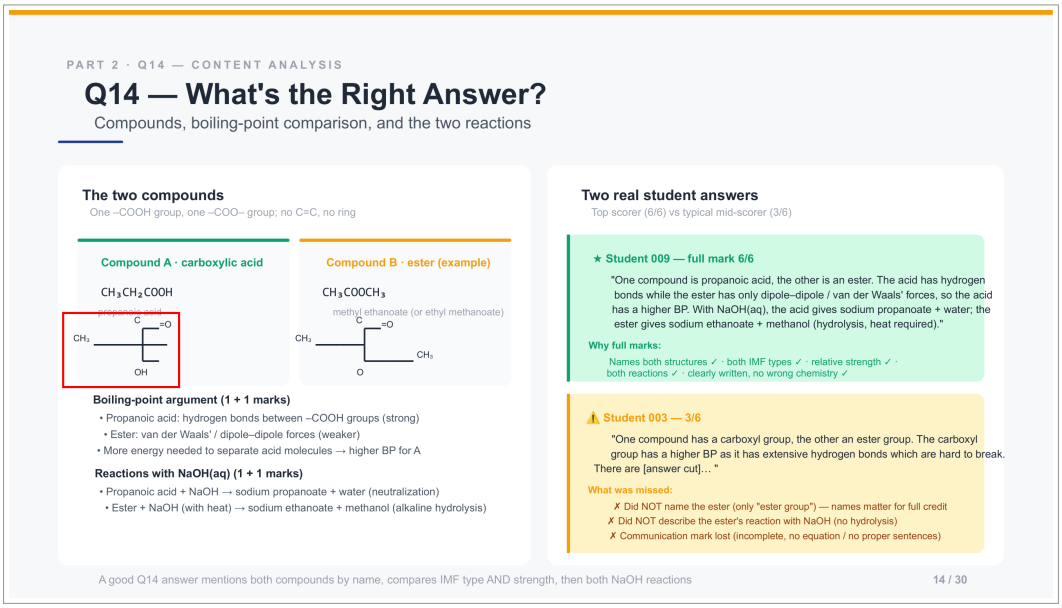


Figure 42: Pattern 5.4: MiniMax M3, Chemistry slides, p. 14. The Compound A drawing connects CH_3 directly to the carboxyl carbon, omitting the intervening CH_2 group required by the adjacent propanoic-acid formula.

H.4.9 Family 6: Missing content or incomplete expression

These patterns concern missing factual information within a presented answer, explanation, or source-based collection. They differ from a wholly missing task deliverable or required component, which is handled by the task completion dimension.

6.1 Truncated content. An expression or passage ends prematurely, leaving its factual meaning incomplete. The missing suffix is essential to the stated answer or claim.

Example (Slides; Biology; GPT-5.5; p. 9; Figure 43). The accepted-answer path says that hydrostatic pressure falls from P to Q "as fluid leaves the." The sentence omits the compartment from which fluid leaves. The rubric explains that water leaves the blood near P, reducing blood volume and hydrostatic pressure along the capillary.

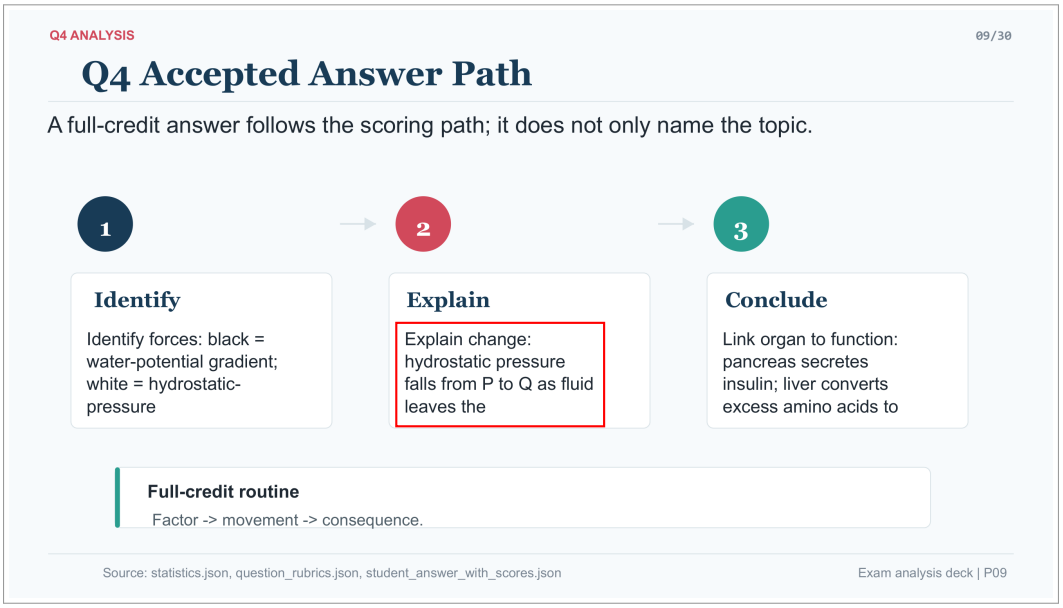


Figure 43: Pattern 6.1: GPT-5.5, Biology slides, p. 9. The answer stops after "as fluid leaves the," omitting the relevant compartment.

6.2 Omitted list items or fields. A collection presented as complete omits a source-required member or field, including entries in a list, table, or chart. The error is the incomplete collection rather than a separately incorrect computed tally.

Example (Student worksheet; Physics; Grok 4.5; p. 4; Figure 44). Student 004's "Marks slipped on" summary for Q1 lists Q1(a), Q1(b), and Q1(c)(i), but omits Q1(c)(ii). The source records the answer "Oven" as $0/1$, also shown in the worksheet's preceding "You wrote" line.

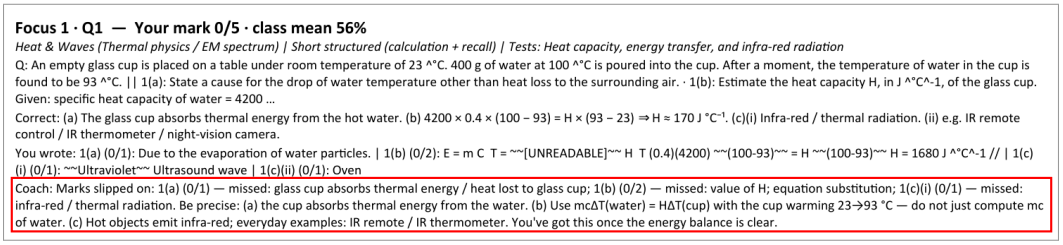


Figure 44: Pattern 6.2: Grok 4.5, Physics student worksheet, p. 4. The boxed missed-mark summary omits Q1(c)(ii), whose $0/1$ result appears immediately above.

6.3 Incomplete cases or alternatives. An answer or explanation omits a valid case when complete coverage is required. Supplying one correct alternative does not make the set complete; explicitly denying another valid case would instead support 4.3.

Example (Slides; Biology; MiniMax M3; p. 12; Figure 45). The Q10(c) accepted-answer summary gives only $50{:}50{:}0$ for homozygous dominant, heterozygous, and homozygous recessive offspring. The rubric also accepts $0{:}50{:}50$. The displayed case is valid, but the accepted-answer coverage is incomplete.

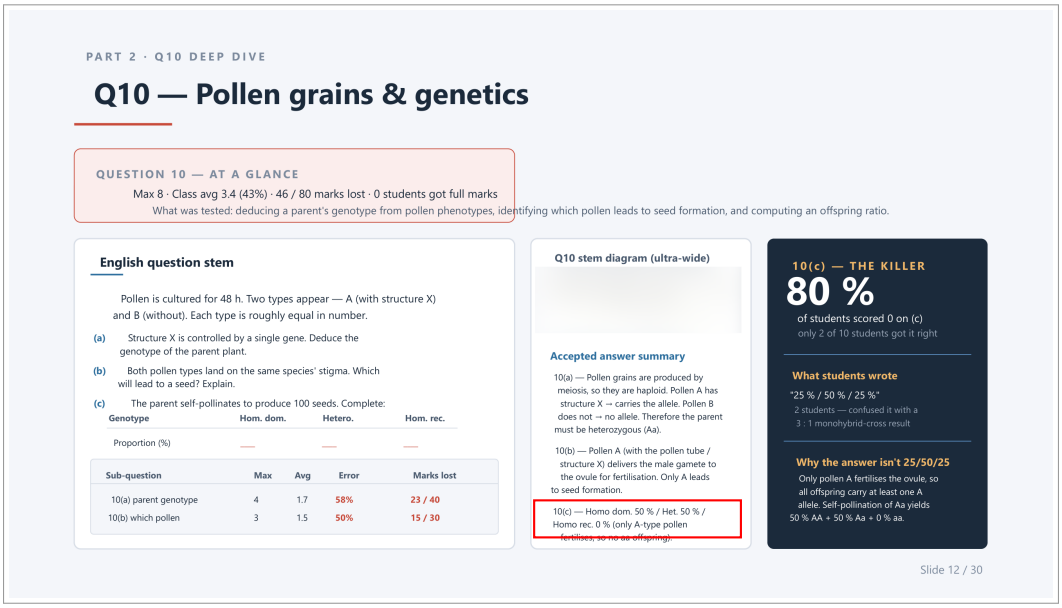


Figure 45: Pattern 6.3: MiniMax M3, Biology slides, p. 12. The accepted-answer summary gives one valid offspring ratio but omits the other accepted ratio. The source exam stem is blurred.

6.4 Missing necessary information or explanatory steps. An answer, argument, or exercise omits an essential condition, element, piece of evidence, reasoning step, or conclusion needed for correctness or completeness. This covers substantive gaps beyond a cut-off string, missing list member, or omitted alternative.

Example (Slides; Chemistry; GPT-5.5; p. 20; Figure 46). The route to full marks for Q11 lists dilution, equilibrium change, K_c, and a graph answer. It omits the required Q11(c) deduction: cooling shifts equilibrium to the reactant side, so the forward reaction is endothermic. The listed route does not cover the full solution.



Figure 46: Pattern 6.4: GPT-5.5, Chemistry slides, p. 20. The route to full marks omits the required Q11(c) equilibrium deduction. The source graph is blurred.

H.4.10 FAMILY 7: INTERNAL DOCUMENT CONTRADICTIONS

These patterns require two identifiable parts of the document that conflict. If the evidence independently establishes a source, domain, or quantitative error, the corresponding earlier family takes precedence. A lack of external exam data alone is not evidence of an internal contradiction.

7.1 Mismatch between self-description and actual content. The document's description of its own coverage, structure, count, sequence, format, or rules is contradicted by the content or instructions it actually provides.

Example (Report; Biology; Claude Opus 5; pp. 1–3). The report says every later action is assigned to evidence-defined subgroups rather than to the whole class. Its Card 1 and Card 2 checks nevertheless request responses from every learner. The instructions contradict the report's description of its own targeting (Figure 47).

Central tendency and spread. Mean 46.1 / 84 (54.9%); median 43.5 (51.8%); standard deviation 18.65 marks (22.2 percentage points); range 21–76; no full marks. Quartiles: p25 31.75, p50 43.5, p75 58.5, p90 75.1. The distribution is wide and thin in the middle — 2 learners in 20–29, 3 in 30–39, one each in 40–49, 50–59 and 60–69, 2 in 70–79. Every action below is therefore assigned to the evidence-defined groups in Part 3 rather than to the class as a whole.

- **Decision test (4 minutes, slips from every learner): A**, no diagram: "Blood flows through a capillary bed. Two different factors act on the fluid across the capillary wall. Name each factor and state which way it drives the fluid." **B**, on the cited script: "student_001 wrote 'fluid enters by osmosis because the water potential inside is lower; fluid leaves by osmosis because the water potential outside is higher'. In one sentence, say what is wrong and correct it." **Expected under the main claim:** A yields one factor only, or water potential named twice; B corrections stay inside osmosis. **Expected under the rival:** A yields both factors with opposite directions (the diagram was the obstacle); B corrections name hydrostatic pressure. **Inspection rule:** A scores 1 only if two distinct named factors appear with opposite directions; tally the class. **Decision:** fewer than 6/10 on A → run the first response below as planned. 6/10 or more on A → the concept is present; replace the re-teach

- **Decision test (5 minutes, slips from every learner):** Put the two cited scripts side by side, anonymised as Script A (student_009, 3/3) and Script B (student_010, 1/3). "Both describe the same Figures II and III. (i) Which claim about the antioxidant's role is supported by the figures? (ii) Name the one feature of the graphs that decides between 'antioxidants raise free radicals' and 'antioxidants suppress free radicals'." **Expected under the main claim:** (i) largely correct once Script A is visible, but (ii) answered by co-variation ("both are high") rather than by the 0–1 h segment. **Expected under the rival:** learners cannot choose in (i) even with both sentences in front of them, having no reason to prefer either mechanism. **Scoring rule:** 1 mark for choosing suppression in (i); 1 mark in (ii) only for naming the 0–1 h segment, in which free radicals stay at the normal level while antioxidants rise. **Decision:** ≥6/10 on (i) but <4/10 on (ii) → teach the segment-and-compare routine below. <6/10 on (i) → teach the antioxidant / free-radical relationship and the general "suppressor" schema first, then re-run this test before spending time on the routine.

Figure 47: Pattern 7.1: Claude Opus 5, Biology report, pp. 1–3. Excerpts show, from top to bottom, the targeting claim (p. 1), Card 1's decision test (p. 2), and Card 2's decision test (p. 3). Both tests request "slips from every learner."

Example (Slides; Mathematics; Claude Opus 5; p. 29; Figure 48). The footer summarizes items 1–4 as covering completion, set-up, part-linking, and "four facts." In the right-hand list, however, "Lock down four one-mark facts" is the fifth item. The list reference therefore excludes one of the recommendations it claims to cover.

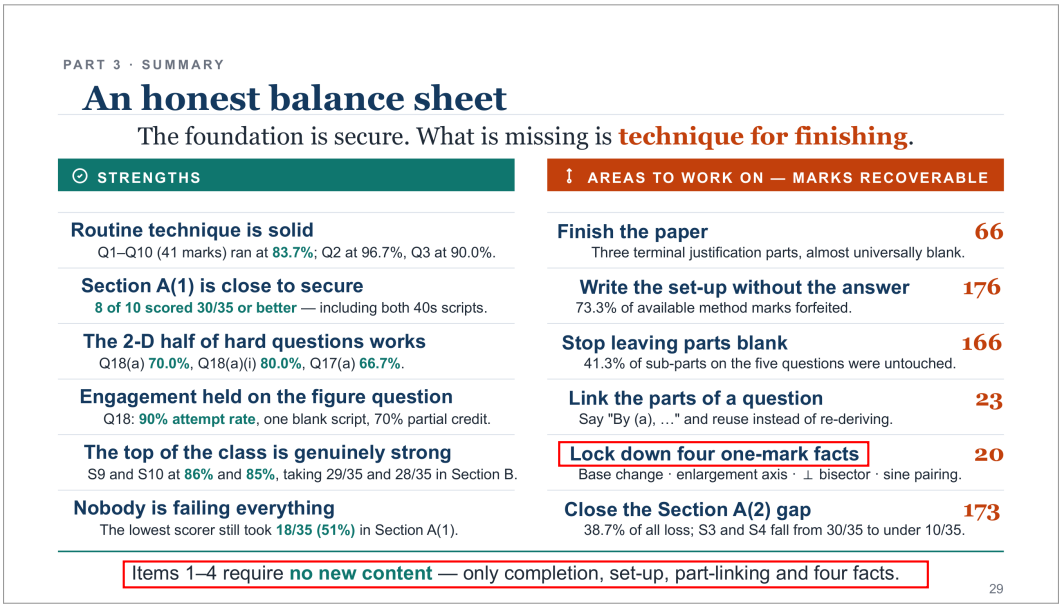


Figure 48: Pattern 7.1: Claude Opus 5, Mathematics slides, p. 29. The footer refers to items 1–4 but includes "four facts," which is the fifth item in the right-hand list.

7.2 Conflicting statements within the document. Two statements about the same matter cannot both hold in the stated context, without the conflict primarily being a description of the document's own structure or format.

I Personalization

I.1 Method and Scoring

I.1.1 Construct and Rubric

Personalization asks whether a document's material educational choices respond to the assessment evidence about its target and are usable for the target's evidenced needs. The target is the individual student for a worksheet and the class for a report or slide deck. Subgroups within a class are required only when the class evidence supports them and the material's purpose would reasonably change. A factually correct explanation can still fail personalization: a worksheet may correctly explain fraction addition when the student needs help with fraction division. Table 19 states the four criteria as evaluation questions (the judge prompt calls them "dimensions").

Table 19: Personalization criteria.

Criterion	Evaluation question
Target alignment	Do the document's choices fit what the student's or class's answers show: the task attempted, the answers given, the content need, the level, the language, and the strengths?
Explanatory grounding	Does the document derive its diagnosis, feedback, and recommendations from that evidence, rather than stating them generically?
Focus	Does the document give the most attention to the needs the evidence shows are most important?
Tailored actionability	Given the chosen need, is the next step a procedure the student or class can use, directed at the right learners, with a check that tests that need?

The pattern catalog in Appendix I.4 illustrates each criterion with cases from the benchmark outputs.

Score anchors. The judge applies the same three anchors to every criterion:

- **0:** The criterion's material choices are essentially generic, target-mismatched, or counterproductive despite adequate target evidence.
- **1:** Target evidence shapes part of the response, but a material genericity, grounding, priority, differentiation, or tailoring gap remains.
- **2:** Target evidence meaningfully shapes the response in this criterion, with at most minor bounded personalization weaknesses.

The prompt gives each criterion its own definition, deduction guidance, and severity anchors for issues; Table 19 summarizes the definitions and our repository provides the full text. Each issue is assigned to one criterion according to its primary repair. A concern is out of scope when its primary repair is a factual correction, an absent required component, a layout repair, or a privacy rule, or when it would not change under different target evidence.

Each judge scores the four criteria in one run and, for every criterion, returns a rationale and up to four evidence-supported issues, each with a low, medium, or high severity.

Issue verdicts. Pattern classification (Appendix I.4) and human review (Appendix I.2) use the following verdict vocabulary: *required*, the issue should be fixed before the document is used; *optional*, a better-tailored choice exists but the document is usable without it, so reporting and omitting it are both acceptable; *noted*, a pattern was assigned for descriptive statistics without endorsing a repair or improvement; *invalid*, the criticism does not hold against the evidence, so reporting it is an error; and *undetermined*, deciding would need task intent or context the inputs do not supply. The classification uses required, optional, noted, and undetermined on grouped occurrences. The human review uses required, optional, invalid, and undetermined on a sample of consolidated claims, where required means the claim should be reported and optional means reporting and omitting it are both acceptable. A judge's severity is one input to the verdict, not the verdict itself. An issue with the required verdict is a **personalization misalignment**; the main text (Section 5.2) counts only misalignments.

Claude Opus 5, GPT-5.5, and Grok 4.5 judge independently. A document is scored only when all three judges return valid records. A deterministic output check requires that the record match the schema and that its scores be consistent with the severity of its reported issues; a record that fails the check is regenerated. We never average fewer than three judgments. Scores are averaged across the three judges within each criterion, and the four criterion means are summed to a 0–8 personalization score; Appendix E.3 defines its contribution to the overall score. The criterion scores remain available to show differences concealed by the total.

Pass thresholds. The pass criterion (Section 4.2) requires a personalization mean of at least 1.67 over the four criteria and a mean of at least 1.33 for each criterion. Both follow from the three-judge, 0–2 scale. A criterion mean of 1.33 is reached when one judge awards a 2 and the other two award a 1, so a criterion clears the floor only when the judges together find it more than partly tailored. The overall threshold is 1.67 rather than $5/3$: over the twelve judgments (three judges, four criteria), exactly two thirds at 2 and the rest at 1 gives 1.667 and does not pass, so strictly more than two thirds must award a 2. For a worksheet run, the same two means are taken over all worksheets in the run (ten, or twenty for English) together with the judges and criteria and compared with the same 1.33 and 1.67, so every task is held to one standard.

I.1.2 Evaluation Context and Reference Records

Each judge receives, separately from the document, a context that states the document type, the intended student or class, the purpose and audience, whether a teacher mediates the material, the privacy and task constraints, and the target's reference records. The reference records are copied directly from the benchmark's structured assessment data. For a report or slide deck, the records are class-level: the exam questions, their maximum marks, and class and question performance statistics, split by examination route where a subject has more than one (English B1 and B2). For a worksheet, the records are student-level: the intended student's answers, marks, and marking rationales; the corresponding questions with answer keys, rubrics, and rationales; and the class statistics

for comparison. Judges work from these records, the prepared document view (Appendix E.2), and the rendered PDF. They are not given factual judgments, defect labels, or agent identity.

I.2 Validation

We compare personalization judging configurations on a selected diagnostic corpus, using human review of pooled judge findings. The comparison assesses the issues judges report; it does not calibrate document-level scores. Two design choices are tested: the *per-criterion* prompt, which scores the four criteria separately, and agentic execution. We ask whether each helps judges identify warranted weaknesses without unwarranted criticism, comparing individual judges and the union of their findings against human-reviewed claims. The Agentic + per-criterion setting with the three judges is the configuration the benchmark uses; a *holistic* prompt, which returns one overall score, and direct API calls serve as comparators.

I.2.1 Experimental Design

The diagnostic corpus contains 18 complete documents: eight slide decks, seven student worksheets, and three class reports, chosen because they raise instructive personalization questions, not to estimate how common issues are. Ten documents use English Language, seven use Biology, and one uses Chemistry. Each document is evaluated under two prompts (holistic and per-criterion), two execution routes (API and agentic), and the same three judges—GPT-5.5, Grok 4.5, and Claude Opus 5—for $18 \times 2 \times 2 \times 3 = 216$ judgments.

Both prompts share the same definition of personalization, the same exclusions, severity definitions, and evidence requirements, and the same rule that one issue is one root cause. The holistic prompt asks for one overall score and up to 16 issues in a single list. The per-criterion prompt asks for one score per criterion and up to four issues per criterion, with the same ceiling of 16. The comparison therefore varies only the criterion guidance and the structure of the output. Our repository provides both prompts.

Both routes receive the same context, text blocks, and expanded text blocks, which describe visual content in words. The context specifies the task constraints and the student or class reference records. The study uses this prepared text view; the benchmark evaluator additionally receives the rendered page images. API judging receives the full context in one request. Agentic judging uses the benchmark's workspace layout and each model's native harness (Codex for GPT-5.5, OpenCode for Grok 4.5, and Claude Code for Claude Opus 5) to read, search, and revisit the files. Both routes are instructed to read the complete document and return one JSON judgment; all 216 judgments yield usable outputs.

I.2.2 Human Reference Claims

GPT-5.5, run through Codex, consolidated the issue entries across all judgments into 211 candidate claims, each preserving its explanations and the judgments that reported it. A claim is one substantive criticism of one educational choice for one target; our repository provides the consolidation procedure.

One author reviewed 72 claims picked at random. The reviewer used LLM-prepared summaries and counterarguments, with source explanations, located evidence, and the original documents available. The review was blind: the interface hid which model, route, and prompt had reported each claim and the machine verdict. Human decisions assign the verdicts of Appendix I.1.1: *required* issues, which should be reported; *optional* improvements, whose reporting and omission are both acceptable; and *invalid* criticisms, whose reporting is an error. Twelve further claims were *undetermined*, mainly because deciding them would require a task intent or information the prompt does not supply. For example, one criticized a worksheet for not explaining why it selected its three focus items. Undetermined claims receive neither credit nor penalty.

Human review yields 33 required issues, 14 optional improvements, 13 invalid criticisms, and 12 undetermined claims. The remaining 60 claims enter the metrics; the 33 required issues span 17 of the 18 documents. The 139 unreviewed candidates enter no metric.

Table 20: Individual-model results on the reviewed claim set. n_{req}, n_{opt}, and n_{inv} are reported counts shown over eligible totals of 33 required issues, 14 optional improvements, and 13 invalid criticisms. Recall R_{req} and supported-finding precision P_{sup} are percentages. Optional omissions are acceptable; undetermined and unreviewed claims enter no metric. Agentic + per-criterion is the configuration the benchmark uses.

Model	Route	Prompt	n_{req}	n_{opt}	n_{inv}	R_{req}	P_{sup}
GPT-5.5	Agentic	Holistic	15/33	1/14	1/13	45.5	94.1
	Agentic	Per-criterion	20/33	0/14	1/13	60.6	95.2
	API	Holistic	16/33	1/14	3/13	48.5	85.0
	API	Per-criterion	24/33	2/14	2/13	72.7	92.9
Grok 4.5	Agentic	Holistic	17/33	0/14	2/13	51.5	89.5
	Agentic	Per-criterion	21/33	0/14	1/13	63.6	95.5
	API	Holistic	16/33	0/14	2/13	48.5	88.9
	API	Per-criterion	17/33	0/14	0/13	51.5	100.0
Claude Opus 5	Agentic	Holistic	20/33	6/14	6/13	60.6	81.2
	Agentic	Per-criterion	25/33	5/14	1/13	75.8	96.8
	API	Holistic	25/33	3/14	8/13	75.8	77.8
	API	Per-criterion	19/33	7/14	2/13	57.6	92.9

Table 21: Union of the same three models' findings within each route and prompt; a claim counts once if any model reports it. Reported counts n_{req}, n_{opt}, and n_{inv} are shown over their eligible totals. Required recall R_{req}, required-only precision P_{req}, and supported-finding precision P_{sup} are percentages. Undetermined and unreviewed claims enter no metric. Bold marks column maxima. Agentic + per-criterion is the configuration the benchmark uses.

Route	Prompt	n_{req}	n_{opt}	n_{inv}	R_{req}	P_{req}	P_{sup}
Agentic	Per-criterion (benchmark)	31/33	5/14	2/13	**93.9**	**93.9**	**94.7**
Agentic	Holistic	23/33	6/14	7/13	69.7	76.7	80.6
API	Per-criterion	27/33	8/14	4/13	81.8	87.1	89.7
API	Holistic	28/33	3/14	11/13	84.8	71.8	73.8

I.2.3 Metrics and Individual-Model Results

Let n_{req}, n_{opt}, and n_{inv} denote the numbers of reported required issues, optional improvements, and invalid criticisms. Required recall R_{req} is the share of the 33 required issues reported. Required-only precision P_{req} is the share of reported required-or-invalid claims that are required, and supported-finding precision P_{sup} the share of all reported metric-eligible claims that are required or optional. Only required omissions count as misses, and only invalid criticisms count as errors; undetermined and unreviewed claims enter no metric.

Table 20 shows incomplete individual coverage: no single judge in any setting detects more than 25 of the 33 required issues. Under agentic execution, all three judges detect more required issues with the per-criterion prompt than with the holistic prompt, with equal or fewer invalid criticisms; the effect of the execution route varies by model. No single judge dominates, which motivates examining their combined coverage.

I.2.4 Combining the Same Three Judges

We next pool the three judges' findings within each route and prompt, using the same judgments as above; no new judgments are made. A claim is counted once if any judge reports it. This matches how the benchmark reports issues: every issue any of the three judges raises. Counting the pooled required, optional, and invalid claims gives n_{req}, n_{opt}, and n_{inv} for the same metrics. For Agentic + per-criterion, these counts are 31, 5, and 2, giving $R_{\text{req}} = 31/33 = 93.9\%$ and $P_{\text{sup}} = 36/38 = 94.7\%$, the figures reported in Appendix E.4. The union pools findings; it neither votes nor averages scores.

On the reviewed claim set, Agentic + per-criterion has the highest required recall and supported precision among the four route–prompt settings (Table 21). Relative to its strongest individual member by detection count, Claude Opus 5, the union raises detections from 25 to 31 and invalid criticisms

from one to two. Thus complementary findings increase coverage while retaining 94.7% supported precision. Compared with API + per-criterion, it detects four more required issues and raises two fewer invalid criticisms. API + per-criterion nevertheless covers more optional improvements (8/14 versus 5/14), and Agentic + holistic covers one more than Agentic + per-criterion (6/14).

Because a union can only add detections and errors, this analysis does not determine an optimal panel size.

I.2.5 LIMITS

Three limits apply. One author reviewed the claims, with assistant-prepared aids. Candidates come from judge outputs, so defects that every judge missed are not represented. GPT-5.5 both consolidated the candidates and served as a judge. The results therefore describe the reviewed set; they do not calibrate the direct scores.

I.3 RESULT DETAILS

Tables 23–25 summarize the 700 reports, slide decks, and worksheets with equal-subject weighting. All 700 documents have valid scores from all three judges. Each table shows all four criterion means (0–2) and their sum (0–8). The criterion means average the three judges, then documents within each subject, and finally the five subjects equally. Summing weights the criteria equally; because strength in one criterion can offset weakness in another, the tables also report each criterion.

Agent tables rank unrounded totals within each task; exact ties share a rank. Displayed values are rounded to two decimals. Table 22 also reports an equal-weight mean across the three tasks.

Table 22: Personalization by task, averaged over three judges, documents within each subject, five equally weighted subjects, and ten agents. Criteria are scored 0–2; Total is their sum (0–8). Higher is better. The final row weights tasks equally. Alignment: target alignment; Grounding: explanatory grounding; Actionability: tailored actionability; Focus uses the full criterion name.

Task	n	Alignment (0–2)	Grounding (0–2)	Focus (0–2)	Actionability (0–2)	Total (0–8)
Reports	50	1.71	1.67	1.55	1.47	6.41
Slides	50	1.79	1.63	1.27	1.15	5.84
Worksheets	600	1.43	1.28	1.82	1.42	5.95
Equal-task mean	—	1.64	1.53	1.55	1.35	6.07

Task profiles. Table 22 supports the task contrasts in Section 5.1. With subjects weighted equally, reports score highest (6.41), then worksheets (5.95), then slides (5.84). The rubric also accounts for how materials are used: slides allow teacher mediation, whereas worksheets must support independent review. Reports require explicit diagnostic tests and teaching responses; their tailored actionability mean is 1.47/2, compared with 1.15/2 for slides.

Agent profiles. GPT-5.5 scores higher overall than GLM 5.2 on worksheets (6.92 versus 6.66), but GLM 5.2 scores higher on explanatory grounding (1.69 versus 1.61). GPT-5.5 scores higher on tailored actionability (1.71 versus 1.52). Overall rankings can therefore conceal different criterion profiles.

Table 23: Reports: personalization by agent (5 reports per agent). Criteria are scored 0–2; Total is their sum (0–8). Higher is better. Rows descend by unrounded total; exact ties share a rank.

Rank	Agent	Alignment (0–2)	Grounding (0–2)	Focus (0–2)	Actionability (0–2)	Total (0–8)
1	GPT-5.5	2.00	1.93	1.73	1.87	7.53
2	Grok 4.5	2.00	1.93	1.53	1.73	7.20
3	Claude Opus 5	2.00	1.80	1.67	1.53	7.00
4	DeepSeek V4 Pro	1.80	1.80	1.67	1.60	6.87
4	GLM 5.2	1.80	1.87	1.67	1.53	6.87
6	Qwen 3.7 Max	1.73	1.73	1.47	1.33	6.27
7	Kimi K2.6	1.73	1.47	1.33	1.40	5.93
8	MiniMax M3	1.33	1.53	1.53	1.40	5.80
9	MiMo V2.5 Pro	1.47	1.27	1.53	1.13	5.40
10	Gemini 3.5 Flash	1.27	1.40	1.33	1.20	5.20

Table 24: Slides: personalization by agent (5 slide decks per agent). Criteria are scored 0–2; Total is their sum (0–8). Higher is better. Rows descend by unrounded total; exact ties share a rank.

Rank	Agent	Alignment (0–2)	Grounding (0–2)	Focus (0–2)	Actionability (0–2)	Total (0–8)
1	Claude Opus 5	1.93	1.87	1.73	1.47	7.00
2	Grok 4.5	2.00	1.87	1.27	1.40	6.53
3	GPT-5.5	1.87	1.87	1.40	1.20	6.33
4	GLM 5.2	1.87	1.67	1.33	1.13	6.00
5	Gemini 3.5 Flash	1.93	1.73	1.07	1.00	5.73
6	DeepSeek V4 Pro	1.73	1.53	1.27	1.00	5.53
6	Kimi K2.6	1.67	1.47	1.33	1.07	5.53
8	Qwen 3.7 Max	1.73	1.53	1.07	1.07	5.40
9	MiniMax M3	1.60	1.47	1.13	1.13	5.33
10	MiMo V2.5 Pro	1.53	1.33	1.13	1.00	5.00

Table 25: Worksheets: personalization by agent (60 worksheets per agent; five subjects weighted equally). Criteria are scored 0–2; Total is their sum (0–8). Higher is better. Rows descend by unrounded total; exact ties share a rank.

Rank	Agent	Alignment (0–2)	Grounding (0–2)	Focus (0–2)	Actionability (0–2)	Total (0–8)
1	Claude Opus 5	1.96	1.98	2.00	1.90	7.84
2	GPT-5.5	1.63	1.61	1.97	1.71	6.92
3	GLM 5.2	1.54	1.69	1.91	1.52	6.66
4	Grok 4.5	1.52	1.41	1.83	1.38	6.14
5	Gemini 3.5 Flash	1.44	1.29	1.83	1.53	6.08
6	MiniMax M3	1.38	1.29	1.74	1.41	5.82
7	Kimi K2.6	1.34	1.08	1.81	1.29	5.51
8	Qwen 3.7 Max	1.21	0.93	1.81	1.24	5.19
9	MiMo V2.5 Pro	1.13	0.79	1.67	1.12	4.71
10	DeepSeek V4 Pro	1.19	0.74	1.63	1.11	4.67

Subject profiles. Table 26 retains the four criteria when splitting results by subject. English Language has the lowest three-judge total in each task, mainly through worksheet explanatory grounding and through report target alignment and tailored actionability. It is not weakest on every criterion: its report focus exceeds the report mean, and Mathematics reports have the lowest report focus.

The English Language class has B1 and B2 routes with different assessment histories; these distinctions can affect diagnosis and allocation. The report cases in Figures 59 and 60 illustrate different route-related mismatches. English also requires 20 worksheets per generation run rather than ten. Biology is not lowest in any task despite its cross-language source grounding. With exam data from

one assessment per subject, these comparisons cannot isolate the effects of exam routes, production demands, or language.

Table 26: Personalization by subject and task across ten agents. Criteria are scored 0–2 and Total 0–8; higher is better. Cells average judges and then documents within the subject–task group. n counts documents, not independent classrooms.

Subject	n	Alignment (0–2)	Grounding (0–2)	Focus (0–2)	Actionability (0–2)	Total (0–8)
Reports						
Biology	10	1.83	1.67	1.60	1.63	6.73
Chemistry	10	1.80	1.73	1.70	1.43	6.67
English	10	1.43	1.67	1.67	1.20	5.97
Mathematics	10	1.70	1.73	1.17	1.50	6.10
Physics	10	1.80	1.57	1.60	1.60	6.57
Slides						
Biology	10	1.93	1.80	1.17	1.23	6.13
Chemistry	10	1.87	1.70	1.37	1.27	6.20
English	10	1.50	1.37	1.13	1.00	5.00
Mathematics	10	1.77	1.63	1.43	1.13	5.97
Physics	10	1.87	1.67	1.27	1.10	5.90
Worksheets						
Biology	100	1.41	1.39	1.74	1.54	6.08
Chemistry	100	1.49	1.50	1.91	1.46	6.36
English	200	1.45	1.02	1.89	1.21	5.57
Mathematics	100	1.37	1.18	1.78	1.45	5.78
Physics	100	1.45	1.30	1.78	1.44	5.97

Selection and diagnosis can separate. The DeepSeek V4 Pro worksheet for English Language Student 001 selects three questions with different errors, yet explains each with the same advice to reread and find keywords, followed by the answer key. All three judges assign explanatory grounding 0, while their focus scores are 1, 2, and 2: two credit the three-item selection, one objects to the prominence of the unranked full error list.

Judge–agent overlap. All three judges also appear among the agents. Including the judge that shares an agent's model, rather than averaging only the other two judges, changes the equal-task total by $+0.27$/8 for GPT-5.5, $+0.09$/8 for Grok 4.5, and -0.12/8 for Claude Opus 5. When each judge is omitted in turn from all agents' scores, the task leaders are unchanged (GPT-5.5 for reports; Claude Opus 5 for slides, worksheets, and the equal-task average), although some lower ranks change. This stability does not establish that self-preference is absent.

I.4 Failure Patterns

This catalog defines the six patterns behind the three families of Section 5.2. The families describe educational choices; they do not replace the four scoring criteria. A factually sound worksheet can still fail personalization, for example by assigning practice that removes the very condition the student got wrong (pattern 2.2 below). Conversely, a misreported answer is a factual error; it earns a personalization penalty only when the feedback then addresses the wrong difficulty.

The counting unit is one issue in one document for one learner or class. The same defect reported by several judges counts once; defective feedback and defective practice for the same question count as two issues. Each issue receives one primary pattern and a verdict (Appendix I.1.1): required, optional, noted, or undetermined. The same defect in two documents or for two learners counts twice, so the counts describe occurrences, not unique misconceptions.

I.4.1 Classification Method

GPT-5.5 groups the judges' issue entries; GPT-6 Astra classifies the resulting issues and then reviews them in a separate pass. Both steps read the student or class evidence and the document content,

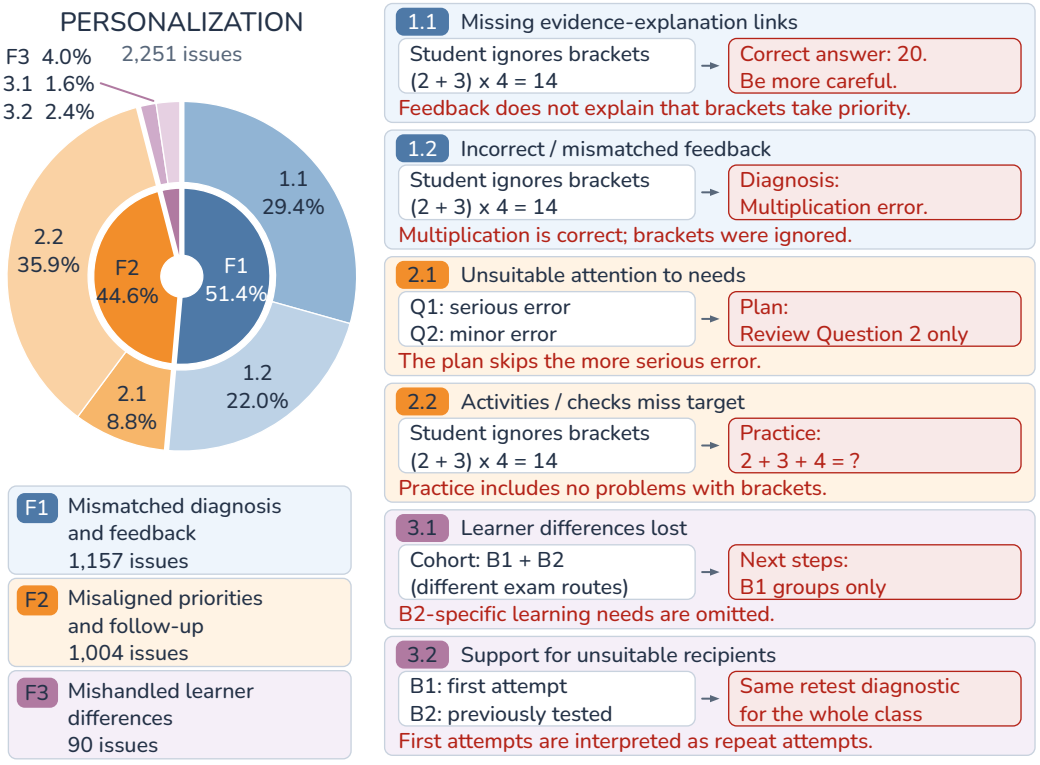


Figure 49: Personalization misalignments in the five-subject main experiment ($n = 2{,}251$). The inner ring shows families and the outer ring shows patterns; percentages use all misalignments. Schematic examples illustrate each pattern.

retrieving further context when needed. The review checks the grouping, the primary pattern, and the verdict. Worksheets are processed per student, so each learner's occurrences stay separate. Labels were then deduplicated, and a human reviewer checked a sample of the assignments, corrected them where needed, and resolved held cases. The classification is descriptive; no held-out accuracy is claimed.

I.4.2 DISTRIBUTION

Overall distribution. The classification covers 700 documents: 50 slide decks, 50 class reports, and 600 student worksheets, across five subjects and ten agents. The 6,744 extracted issue entries were grouped into 4,193 occurrences, with verdicts of required for 2,251, optional for 1,321, noted for 565, and undetermined for 56 (Table 32).

The primary tables use the 2,251 required occurrences, the personalization misalignments analyzed in Section 5.2. Mismatched diagnosis and feedback account for 1,157 (51.4%), misdirected priorities and follow-up for 1,004 (44.6%), and mishandled learner differences for 90 (4.0%) (Table 27). Percentages use the required occurrences in the corresponding task, subject, or agent as their denominator. The pooled column counts occurrences, so worksheets dominate it with 1,917 of the 2,251 (85.2%). Optional improvements are examined separately below.

Table 27: Personalization failure families among misalignments (required verdict). Counts represent grouped issues requiring repair; percentages show each family's share of such issues for that task or for all tasks combined.

Family	Slides	Reports	Worksheets	All
1. Mismatched diagnosis and feedback	59 (37.8%)	34 (19.1%)	1,064 (55.5%)	1,157 (51.4%)
2. Misdirected priorities and follow-up	73 (46.8%)	85 (47.8%)	846 (44.1%)	1,004 (44.6%)
3. Mishandled learner differences	24 (15.4%)	59 (33.1%)	7 (0.4%)	90 (4.0%)
Misalignments	156 (100.0%)	178 (100.0%)	1,917 (100.0%)	2,251 (100.0%)

Results by task and subject. Table 28 provides the supporting task–subject breakdown. The document counts include outputs with no required occurrence. Table 29 gives the document-level view: 650 of the 700 documents (92.9%) contain at least one required occurrence.

Table 28: Required personalization occurrences by task and subject.

Subject	N	F1	F2	F3	Total
Slides					
Biology	10	5	16	1	22
Chemistry	10	12	5	2	19
English	10	12	11	6	29
Mathematics	10	17	18	6	41
Physics	10	13	23	9	45
Slides total	50	59	73	24	156
Reports					
Biology	10	4	9	9	22
Chemistry	10	8	22	11	41
English	10	5	8	14	27
Mathematics	10	7	20	13	40
Physics	10	10	26	12	48
Reports total	50	34	85	59	178
Worksheets					
Biology	100	179	116	0	295
Chemistry	100	163	130	6	299
English	200	318	233	0	551
Mathematics	100	180	197	0	377
Physics	100	224	170	1	395
Worksheets total	600	1064	846	7	1917
All	700	1157	1004	90	2251

N counts all evaluated documents, including those with no required occurrence. F1: mismatched diagnosis and feedback; F2: misdirected priorities and follow-up; F3: mishandled learner differences. Compare family composition within rows; raw occurrence counts are not quality rankings.

Table 29: Documents with at least one personalization misalignment (required verdict), by task.

Task	**Documents**	**With $\geq$1 misalignment**	**Share**
Slides	50	46	92.0%
Reports	50	47	94.0%
Worksheets	600	557	92.8%
All	700	650	92.9%

Differences between tasks. Misdirected priorities and follow-up are the largest family in both slides and reports, while mismatched diagnosis and feedback dominate worksheets. Cases of mishandled learner differences are concentrated in reports and remain rare in worksheets, consistent with a worksheet's single-learner target, which leaves few learner distinctions to lose or misassign.

Differences between subjects. Misdirected priorities and follow-up lead slides in Biology, Mathematics, and Physics, while mismatched diagnosis and feedback lead in Chemistry and English. In reports, misdirected priorities lead or tie in four subjects, but English reports are led by mishandled learner differences, the B1 and B2 route problem shown in the examples below. In worksheets, mismatched diagnosis and feedback lead in every subject except Mathematics, where activity mismatch edges ahead. English has twice as many worksheets as the other subjects, so raw totals do not compare subject difficulty.

Results by task and agent. Table 30 gives a supporting breakdown by agent.

Table 30: Required personalization occurrences by task and producing agent.

Agent (harness / model)	N	F1	F2	F3	Total
Slides					
Codex / GPT-5.5	5	3	4	3	10
OpenCode / Grok 4.5	5	4	5	3	12
Claude Code / Claude Opus 5	5	3	10	2	15
OpenCode / Gemini 3.5 Flash	5	5	9	3	17
OpenCode / GLM 5.2	5	6	5	3	14
OpenCode / MiMo V2.5 Pro	5	4	3	1	8
OpenCode / Kimi K2.6	5	9	6	2	17
OpenCode / DeepSeek V4 Pro	5	8	8	4	20
OpenCode / Qwen 3.7 Max	5	7	11	2	20
OpenCode / MiniMax M3	5	10	12	1	23
Slides total	50	59	73	24	156
Reports					
Codex / GPT-5.5	5	3	4	1	8
OpenCode / Grok 4.5	5	1	8	3	12
Claude Code / Claude Opus 5	5	1	6	6	13
OpenCode / Gemini 3.5 Flash	5	8	9	7	24
OpenCode / GLM 5.2	5	0	9	6	15
OpenCode / MiMo V2.5 Pro	5	5	13	7	25
OpenCode / Kimi K2.6	5	5	10	8	23
OpenCode / DeepSeek V4 Pro	5	4	4	6	14
OpenCode / Qwen 3.7 Max	5	2	8	7	17
OpenCode / MiniMax M3	5	5	14	8	27
Reports total	50	34	85	59	178
Worksheets					
Codex / GPT-5.5	60	123	50	0	173
OpenCode / Grok 4.5	60	110	76	0	186
Claude Code / Claude Opus 5	60	35	30	6	71
OpenCode / Gemini 3.5 Flash	60	79	95	0	174
OpenCode / GLM 5.2	60	80	68	0	148
OpenCode / MiMo V2.5 Pro	60	130	124	0	254
OpenCode / Kimi K2.6	60	133	111	0	244
OpenCode / DeepSeek V4 Pro	60	122	104	0	226
OpenCode / Qwen 3.7 Max	60	132	104	0	236
OpenCode / MiniMax M3	60	120	84	1	205
Worksheets total	600	1064	846	7	1917
All	700	1157	1004	90	2251

N counts all evaluated documents, including those with no required occurrence. F1: mismatched diagnosis and feedback; F2: misdirected priorities and follow-up; F3: mishandled learner differences. Compare family composition within rows; raw occurrence counts are not quality rankings.

The agent profiles broadly follow the task-level family distribution, but the small slide and report denominators make fine-grained comparisons unstable. Claude Opus 5 has the fewest worksheet occurrences (71) and MiMo V2.5 Pro the most (254). Six of the seven worksheet occurrences of mishandled learner differences are Claude Opus 5's; the seventh is MiniMax M3's.

Detailed pattern counts. Table 31 assigns each required occurrence to exactly one of the six patterns. Activities or success checks that do not fit the target (2.2) and missing evidence-to-explanation connections (1.1) are the two largest patterns. Within Family 2, slide occurrences are mostly attention problems (2.1) and worksheet occurrences mostly activity problems (2.2); within Family 3, report occurrences chiefly concern recipients (3.2) and slide occurrences chiefly lost distinctions (3.1). The examples below illustrate both patterns in each family.

Table 31: Personalization patterns among misalignments (required verdict).

Pattern	Slides	Reports	Worksheets	All
1.1 Missing evidence-to-explanation connections	18	18	625	661
1.2 Incorrect or mismatched diagnosis, explanation, or advice	41	16	439	496
2.1 Needs receive unsuitable attention	45	36	116	197
2.2 Activities or success checks do not fit the target	28	49	730	807
3.1 Relevant learner differences are lost	21	16	0	37
3.2 Support is assigned to unsuitable recipients	3	43	7	53
Total	156	178	1,917	2,251

Each required occurrence contributes to exactly one pattern row. The six cases in the pattern catalog illustrate these patterns.

Verdicts and sensitivity to optional improvements. Table 32 reports all 4,193 occurrences by verdict. Noted and undetermined occurrences are excluded from the family distributions; neither is a false-positive count.

Table 32: Verdicts of all grouped personalization occurrences.

Verdict	Slides	Reports	Worksheets	All
Required (misalignments)	156	178	1,917	2,251
Optional	137	81	1,103	1,321
Noted	75	88	402	565
Undetermined	16	19	21	56
Total	384	366	3,443	4,193

Noted means that a pattern was assigned without endorsing a repair or improvement; it is not a false-positive count. Verdicts are defined in Appendix I.1.1. The document denominators are 50 slide decks, 50 reports and 600 worksheets.

Including optional improvements changes the pooled ranking: misdirected priorities and follow-up become the largest family (Table 33). The lead of mismatched diagnosis and feedback therefore holds for misalignments, not for all potentially useful improvements.

Table 33: Personalization family composition when optional improvements are included.

Family	Slides	Reports	Worksheets	All
1 Mismatched diagnosis and feedback	73 (24.9%)	46 (17.8%)	1,298 (43.0%)	1,417 (39.7%)
2 Misdirected priorities and follow-up	165 (56.3%)	146 (56.4%)	1,715 (56.8%)	2,026 (56.7%)
3 Mishandled learner differences	55 (18.8%)	67 (25.9%)	7 (0.2%)	129 (3.6%)
Total	293	259	3,020	3,572

Percentages use required-plus-optional occurrences in each column. Noted and undetermined rows are excluded.

I.4.3 Pattern Catalog

Definitions and illustrative examples. The following catalog gives one misalignment example for each pattern. These six occurrences come from five documents: patterns 1.2 and 2.2 concern different defects in the same worksheet. The cases were selected for clarity. Each identifies the learner evidence, the generated response, and the supported mismatch, and notes counterevidence where relevant. Learner-side evidence (answers, marks, and route membership) is quoted from the benchmark's assessment records; the crops reproduce the generated document.

Slide examples retain the full rendered page; worksheet and report examples use crops with enough surrounding content to interpret the marked passage. Red boxes identify the relevant evidence; not every enclosed sentence is wrong. Original student work is explicitly distinguished from generated material. Worksheet page numbers refer to the named student's individual worksheet; report and

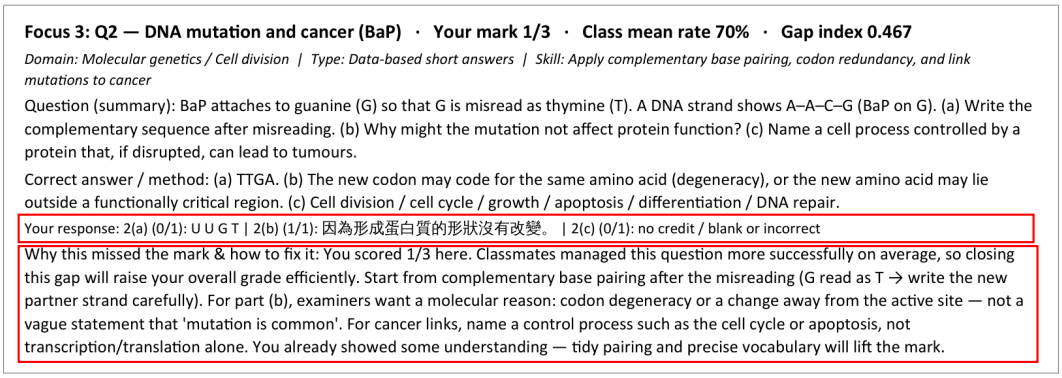
slide page numbers refer to the source document. Student identifiers are local to each subject's class.

I.4.4 Family 1: Mismatched diagnosis and feedback

This family concerns how the teaching material explains a learner's performance and what corrective advice it gives. Pattern 1.1 leaves a needed connection to the evidence unexplained; pattern 1.2 supplies an incorrect, unsupported or mismatched account.

1.1 Missing evidence-to-explanation connections. Relevant learner or task evidence is available, but the explanation does not connect it to the learner's answer, lost credit or needed correction. Presenting the correct answer alone may leave the learner's error unexplained.

Example (Student worksheet; Biology; Grok 4.5; Student 004; p. 2). Q2(a) asks for a complementary DNA sequence under a base-misreading condition. The student wrote `U U G T`, earning 0/1; the required answer is `TTGA`. The worksheet prints both answers but only advises careful complementary pairing. It never explains that DNA uses thymine (T), whereas uracil (U) belongs to RNA, leaving an observable error in this student's answer unaddressed.



Focus 3: Q2 — DNA mutation and cancer (BaP) · Your mark 1/3 · Class mean rate 70% · Gap index 0.467

Domain: Molecular genetics / Cell division | Type: Data-based short answers | Skill: Apply complementary base pairing, codon redundancy, and link mutations to cancer

Question (summary): BaP attaches to guanine (G) so that G is misread as thymine (T). A DNA strand shows A–A–C–G (BaP on G). (a) Write the complementary sequence after misreading. (b) Why might the mutation not affect protein function? (c) Name a cell process controlled by a protein that, if disrupted, can lead to tumours.

Correct answer / method: (a) TTGA. (b) The new codon may code for the same amino acid (degeneracy), or the new amino acid may lie outside a functionally critical region. (c) Cell division / cell cycle / growth / apoptosis / differentiation / DNA repair.

Your response: 2(a) (0/1): U U G T | 2(b) (1/1): 因為形成蛋白質的形狀沒有改變。 | 2(c) (0/1): no credit / blank or incorrect

Why this missed the mark & how to fix it: You scored 1/3 here. Classmates managed this question more successfully on average, so closing this gap will raise your overall grade efficiently. Start from complementary base pairing after the misreading (G read as T → write the new partner strand carefully). For part (b), examiners want a molecular reason: codon degeneracy or a change away from the active site — not a vague statement that 'mutation is common'. For cancer links, name a control process such as the cell cycle or apoptosis, not transcription/translation alone. You already showed some understanding — tidy pairing and precise vocabulary will lift the mark.

Figure 50: Pattern 1.1: Grok 4.5, Biology worksheet, Student 004, p. 2. The complete Focus 3 block shows the student's answer and the explanation that leaves its U-for-T error unnamed.

1.2 Incorrect or mismatched diagnosis, explanation, or advice. The offered diagnosis, explanation or corrective guidance is incorrect, unsupported by the learner evidence, or directed at a different need. The problem lies in what the feedback says, rather than a missing explanation.

Example (Student worksheet; Biology; Claude Opus 5; Student 007; pp. 1–2). The student's record shows liver for the insulin row and kidney for the urea row, both without credit; the crops reproduce only the insulin row. The correct organs are pancreas and liver, respectively. The worksheet suggests that the student swapped the two rows and emphasizes checking their order. A swap would have put pancreas in the urea row, so the student's kidney answer does not support that diagnosis. The worksheet does teach the correct organs elsewhere; the failure is its unsupported explanation of this learner's errors. Figures 51 and 52 show the diagnosis across both pages.

FOCUS 2 — QUESTION 4(c)(i) your mark 0 / 1 class average on this question 80%	
The question	Q4(c) gave a table about capillary networks in which substances are absorbed into the blood. In the row you had to complete, the substance absorbed was insulin and the explanation given was: "the organ responds to a change in blood glucose level and secretes insulin." Name the organ.
What you wrote	*"Liver."*
Correct answer / method	The pancreas.
Why it went wrong, and how to fix it	Liver is the correct organ for the **urea** row just underneath, so it looks as though the two rows got swapped. The explanation printed beside this blank says the organ "responds to a change in blood glucose level

Figure 51: Pattern 1.2: Claude Opus 5, Biology worksheet, Student 007, p. 1. Focus 2 proposes a row swap; its explanation continues on the next source page.

	and secretes insulin" — that is the **pancreas**. On a paper as strong as yours these are the cheapest marks to win back: read the explanation column first, write the organ that matches it, then check each row against its own explanation before moving on.
Topic · type · skill	**Domain:** Organisms and Environment — endocrine organs and regulation of blood glucose **Type:** Table completion; recall in context (1 mark) **Tests:** Matching a substance to the organ that puts it into the blood — insulin and glucagon from the pancreas, urea from the liver.

Figure 52: Pattern 1.2: Claude Opus 5, Biology worksheet, Student 007, p. 2. Continuation of Focus 2 and its row-checking advice.

I.4.5 Family 2: Misdirected priorities and follow-up

This family concerns which learning needs receive attention and how those needs are practiced or checked. Pattern 2.1 concerns the selection or emphasis of needs; pattern 2.2 concerns the activity assigned to an already selected need.

2.1 Needs receive unsuitable attention. A need is selected, emphasized, omitted or displaced in a way that does not fit the evidence and task constraints, or an identified priority is not carried through into the plan. The issue is which need receives attention.

Example (Slides; English Language; DeepSeek V4 Pro; pp. 20, 24). The deck identifies text-to-structure mapping—flow charts, paragraph matching and heading matching—as the B1 learners' first priority. Its main four-week plan instead schedules vocabulary, causal reasoning, reference identification and integrated practice, without a named slot for that priority.

Slide 26 prescribes daily flow-chart and paragraph-matching practice, so the priority is not absent from the deck. The gap is in the weekly plan: it gives the declared first priority no slot and no success check, and its final whole-paper score does not isolate that skill, so a class following the plan never has its largest evidenced need tested on its own. Figures 53 and 54 show the stated priority and weekly plan.

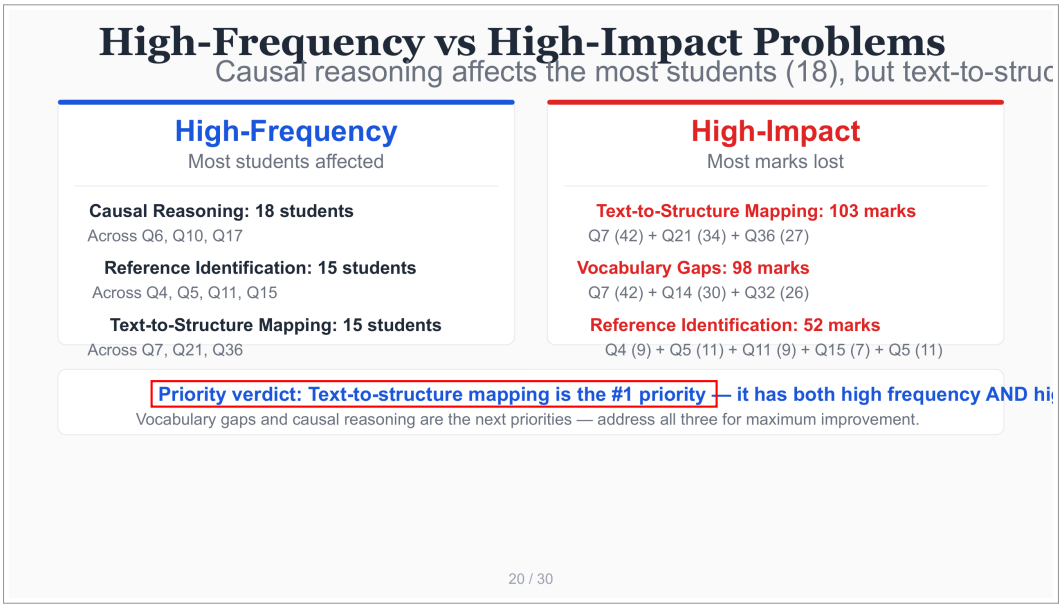


Figure 53: Pattern 2.1: DeepSeek V4 Pro, English Language slides, p. 20. Text-to-structure mapping is declared the "#1 priority."

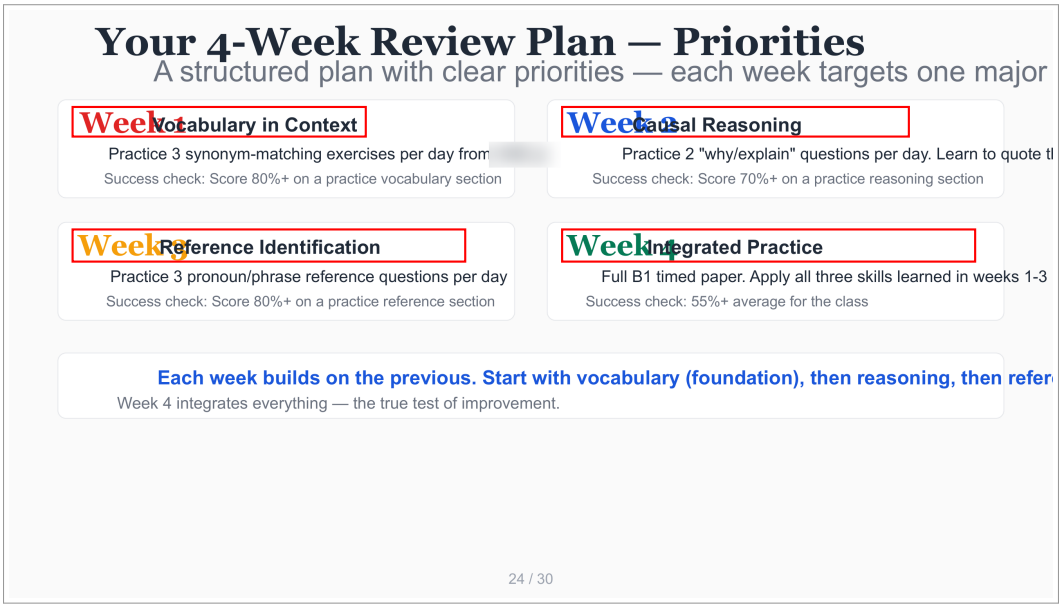


Figure 54: Pattern 2.1: DeepSeek V4 Pro, English Language slides, p. 24. The full weekly plan has no named slot for text-to-structure mapping. Original overlap and clipping are retained.

2.2 Activities or success checks do not fit the target. An assigned exercise, support procedure or success criterion does not address the selected learning target. It may bypass the needed skill or remove a condition that made the original answer difficult.

Example (Student worksheet; Biology; Claude Opus 5; Student 007; pp. 1–2). Q6(c) asks the student to modify a product–time curve when an enzyme inhibitor is added at time t. The curve must remain unchanged before t and lie below the original afterwards. According to the marking record,

the student drew a roughly straight rising line that is already below the original before t, giving the inhibitor an effect before it is added; the answer received 0/1.

The worksheet correctly explains this timing condition, but its matched practice asks about enzyme solution boiled *before the reaction starts*. That scenario calls for little or no product throughout and removes the need to preserve a normal pre-t segment. The practice therefore bypasses the specific error visible in the student's drawing, despite the correct feedback. Figure 55 is a reconstruction of the timing error from that record, not the student's script; Figures 56 and 57 show the worksheet's feedback and practice.

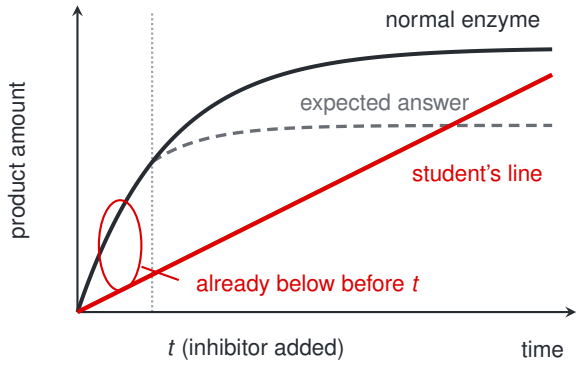
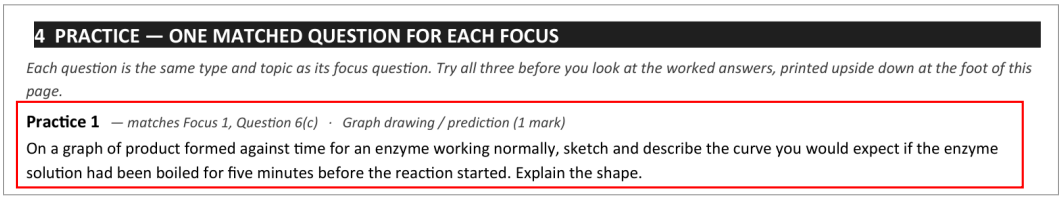


Figure 55: Pattern 2.2: reconstruction of the timing error from the marking record, not the student's script. The student's line is already below the normal-enzyme curve before the inhibitor is added at t; the expected answer diverges only afterwards.

FOCUS 1 — QUESTION 6(c) your mark 0 / 1 class average on this question 90%	
The question	Q6(c) showed a graph of the amount of product formed against time while the amylase was working normally. The experiment was repeated and bean extract was added at time t. If the bean extract inhibits the amylase, draw a line on the graph to show the result.
What you wrote	*A line was drawn on the graph, but it did not show the effect of an inhibitor.*
Correct answer / method	The line must lie exactly on the original curve up to time t, and after t it must rise more gently than the original (it may level off) and stay below it — it must never come down.
Why it went wrong, and how to fix it	The line you drew was on the graph but the shape did not match what an inhibitor does. The marker wants a line that lies **exactly on the original curve up to time t** — nothing has changed yet — and then, from t onwards, **rises less steeply and stays below** the original curve; it may flatten out, but it must **never come down**, because the product already made does not disappear. Before you draw, say the idea to yourself: "inhibitor = slower rate = smaller gradient, starting at t."
Topic · type · skill	**Domain:** Cells and Molecules of Life — enzymes and enzyme inhibitors **Type:** Graph drawing / prediction (1 mark) **Tests:** Showing that an inhibitor reduces the rate of product formation from the moment it is added — a smaller gradient after t — while the product already made does not disappear.

Figure 56: Pattern 2.2: Claude Opus 5, Biology worksheet, Student 007, p. 1. Focus 1 correctly explains that the curve should remain unchanged up to t.



4 PRACTICE — ONE MATCHED QUESTION FOR EACH FOCUS

Each question is the same type and topic as its focus question. Try all three before you look at the worked answers, printed upside down at the foot of this page.

Practice 1 *— matches Focus 1, Question 6(c) · Graph drawing / prediction (1 mark)*

On a graph of product formed against time for an enzyme working normally, sketch and describe the curve you would expect if the enzyme solution had been boiled for five minutes before the reaction started. Explain the shape.

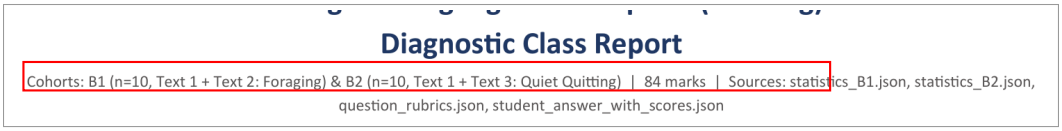
Figure 57: Pattern 2.2: Claude Opus 5, Biology worksheet, Student 007, p. 2. Practice 1 is explicitly matched to Q6(c), but boiling the enzyme before the reaction removes the timing condition.

I.4.6 Family 3: Mishandled learner differences

This family concerns whose needs are represented and who receives support. Pattern 3.1 loses a relevant group or distinction from the plan; pattern 3.2 assigns a specified intervention to learners for whom that use is unsupported. Shared instruction is appropriate when it fits the evidence. In the English examples below, B1 and B2 are examination routes with different Part B questions.

3.1 Relevant learner differences are lost. A relevant subgroup, assessment route or need profile is omitted or pooled into guidance that fails to preserve the needed distinction. The failure concerns missing coverage or differentiation in the planned support.

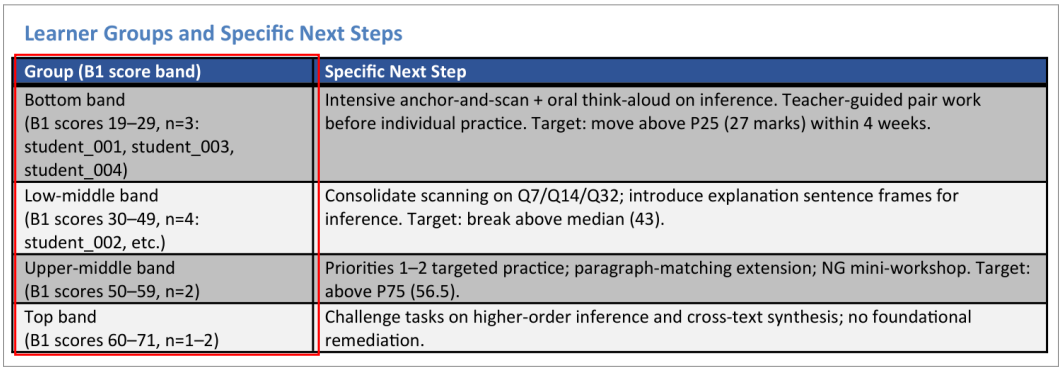
Example (Report; English Language; Qwen 3.7 Max; pp. 1, 4). The report includes ten B1 and ten B2 learners in its declared cohort, but its learner-group table gives next steps only to B1 groups. Its priorities and diagnostic targets likewise use B1 evidence. B2 learners have distinct needs: the class records show all ten scoring zero on their route-specific Q56 referent question, yet the plan provides no corresponding B2 priority or next step. Shared vocabulary or inference teaching may still benefit them; the failure is leaving their distinct needs outside the instructional plan. Figures 58 and 59 show the declared cohort and learner-group plan.



Diagnostic Class Report

Cohorts: B1 (n=10, Text 1 + Text 2: Foraging) & B2 (n=10, Text 1 + Text 3: Quiet Quitting) | 84 marks | Sources: statistics_B1.json, statistics_B2.json, question_rubrics.json, student_answer_with_scores.json

Figure 58: Pattern 3.1: Qwen 3.7 Max, English Language report, p. 1. The declared cohort includes both ten-learner routes.



Learner Groups and Specific Next Steps

Group (B1 score band)	Specific Next Step
Bottom band (B1 scores 19–29, n=3: student_001, student_003, student_004)	Intensive anchor-and-scan + oral think-aloud on inference. Teacher-guided pair work before individual practice. Target: move above P25 (27 marks) within 4 weeks.
Low-middle band (B1 scores 30–49, n=4: student_002, etc.)	Consolidate scanning on Q7/Q14/Q32; introduce explanation sentence frames for inference. Target: break above median (43).
Upper-middle band (B1 scores 50–59, n=2)	Priorities 1–2 targeted practice; paragraph-matching extension; NG mini-workshop. Target: above P75 (56.5).
Top band (B1 scores 60–71, n=1–2)	Challenge tasks on higher-order inference and cross-text synthesis; no foundational remediation.

Figure 59: Pattern 3.1: Qwen 3.7 Max, English Language report, p. 4. The complete learner-group table gives next steps only for B1 score bands.

3.2 Support is assigned to unsuitable recipients. An intervention is assigned using a recipient list or selection rule that does not match the learners for whom its use is supported. The failure concerns who receives the intervention or how their results are interpreted.

Example (Report; English Language; Claude Opus 5; p. 3). The class contains ten B1 and ten B2 learners. Only the B2 route's paper contains Q51 and Q61, so only B2 previously sat them, but Card B assigns both questions to the whole class, calls them "already-sat," and removes time pressure to distinguish skill difficulty from exam-condition effects. For B1, this is a first assessment: their answers cannot show whether removing time pressure resolves a previous failure on these items. Supplying the reading passages makes the questions answerable, but does not supply the prior assessment required for that comparison.

The predictions explicitly use B2 counts, and later instructions start B1 on Q4/Q15. Those adaptations narrow the criticism, but the whole-class probe still gives no separate interpretation for B1's first encounter. The failure is assigning this repeated-item diagnostic across different assessment histories; shared new practice would not itself be inappropriate.

- **Decision test (5 minutes, whole class, one slip each).** With the paragraphs available and no time pressure, each learner writes, for two already-sat referring expressions, **(a)** the referent and **(b)** the 3–5 words in the paragraph that prove it. *Control:* Q51 'it' (line 21), explicit antecedent. *Target:* Q61 'forebears' (line 50), description-defined. Then: of three answers this class gave to Q61 — 'The old generation' (student_014), 'The senior workers' (student_017), 'Millennials' (student_020) — which does the paragraph support, and which words show it? *Main claim predicts:* control near ceiling (≥8/10 of B2-route scripts on Q51) but target still weak (<6/10 on Q61), proof phrases missing or generic. *Rival predicts:* removing time pressure suffices — Q61 correct on ≥8/10, proof phrases cited. *Rule:* score (a) and (b) separately; compare control against target *within* each script. *Decisions:* skill result → the first response below, both route groups. Position/text result → do not re-teach reference; spend the time on pacing and order of attempt, and re-check with a referent item placed late in a timed section.

Figure 60: Pattern 3.2: Claude Opus 5, English Language report, p. 3. Card B assigns the "already-sat" diagnostic to the whole class, although only B2 previously sat these items.

J Detailed Experimental Setup and Generation Prompts

This appendix details the generation setup and presents the producer prompts for the class-report, slide, and student-worksheet tasks; each defines what the producing agent is asked to deliver. The prompts used to evaluate the delivered materials are provided in our repository (see the end of this appendix). Each prompt is used unchanged for all five subjects, and the three tasks receive the same exam data for a given subject. The exam data, which vary by subject, and the available skills, which vary by task, are supplied alongside the prompt and are not reproduced here.

Agent setup. Each agent works in a workspace containing the input data, receives a prompt with the task instructions, and has access to a shared set of skills. All runs are executed by the multi-agent framework described in Section 3. Each run allows up to three generation attempts.

Input data. In each run, an agent performs one task for one subject (Section 2) and receives four inputs in its workspace: `question_rubrics.json`, containing the questions, expected answers, and grading rubrics; `statistics.json`, containing class- and question-level performance summaries; `student_answer_with_scores.json`, containing each learner's answers, awarded scores, and scoring rationales; and an image directory containing non-textual information such as charts and diagrams.

Instructions. The producer prompts specifying each task and deliverable are provided in Subsection J.2.

Skills. All agents share three skills packaged with `SKILL.md` instructions (OpenAI, 2026b) that provide required tools and standardized common subtasks: `latex-formula` renders LaTeX formulas as images for insertion into documents; `uv-python` runs Python code through `uv`; and `vision-delegate` lets agents without native vision send images to GPT-5.4 mini via API and receive text descriptions, while agents with native vision inspect images directly. Slide tasks also use `ppt-master` (He, 2026) to build PPTX slide decks. Subsection J.1 details each skill and the tasks that use it.

Terminology in reproduced prompts. We call the individual feedback-and-practice documents *personalized worksheets*, abbreviated to *Worksheets* in compact tables; the appendix also writes *student worksheets* where the per-student file is the point. The generation instructions use "student sheets" for the same documents, and the implementation identifier `student_sheets`, which appears in some prompts and checklists, refers to the same task.

Instructions for alternative routes. The prompts contain general instructions for handling alternative sections, without naming English reading's B1/B2 groups or prescribing an English-specific teaching plan. The report prompt requires identifying the section, part, or optional route covered by the evidence and defining learner groups from supplied evidence. The slide prompt has no explicit route-handling instruction. The student-worksheet prompt requires identifying the learner's route and comparing that learner only with students who sat the same section. Thus agents must interpret the supplied route evidence and determine its consequences for diagnosis, priorities, and follow-up.

J.1 Agent Skills

The generation pipeline exposes the four skills below as directories with `SKILL.md` instructions and `name` and `description` metadata, following the skill structure documented by OpenAI (OpenAI, 2026b). All three tasks receive `latex-formula`, `uv-python`, and `vision-delegate`; the slides task also receives `ppt-master`. The available skills are supplied separately from the task prompts reproduced below.

`latex-formula`
Provides a script that renders formula-only LaTeX as PNG, SVG, or PDF images with Matplotlib mathtext, without a system TeX installation, so that formulas can be inserted into documents (`skills/utils/latex-formula/SKILL.md`).

`uv-python`
Guides Python execution, environment setup, and dependency management through `uv`; it bundles no script (`skills/utils/uv-python/SKILL.md`).

`vision-delegate`
Directs agents with native vision to inspect images directly. Agents without native vision instead use a bundled script that sends one image and a task prompt to a vision-language model, GPT-5.4 mini in our runs, and returns its text description (`skills/utils/vision-delegate/SKILL.md`).

`ppt-master`
Provides the PPTX creation workflow for the slides task (`skills/utils/slides/ppt-master/SKILL.md`) (He, 2026).

J.2 Producer Prompts

J.2.1 Class-Level Report Creation

Listing 1 gives the class-report prompt; Section 2 summarizes its requirements.

Listing 1: Producer prompt for class-level report creation.

```
# Identity

You are an expert senior-high-school teacher, assessment analyst, and document
designer. You create compact, evidence-constrained class reports that help a
teaching colleague decide what to test and teach next.

# Instruction

Create a diagnostic class report from one class assessment.

1. Read every file in the `source_materials` folder.
2. Write the report in English as `class_report.docx` in the workspace root and
   export the same document as `class_report.pdf` using LibreOffice headless.
3. Keep the complete report, including title and source locators, within four A4
   pages, using body text of at least 10.5 pt and margins of at least 18 mm. Do
   not use compressed typography, unreadably dense tables, or off-page content.
4. You are working in an automated pipeline and may proceed without asking for
   further confirmation.
```

```
Use only supplied evidence. Do not invent scores, response patterns,
quotations, learner groups, marking requirements, exemplars, or locators.
Distinguish observation from inference, and narrow or qualify claims when the
evidence is insufficient.

The report should contain three main parts:

## 1. Cohort context and ranked diagnostic priorities

- Briefly establish the cohort context using the supplied numeric central
  tendency and spread. Identify the section, part, or optional route covered by
  the evidence, or state that the paper offers none.
- Answer what is breaking down, why, what should be re-taught first, what rival
  explanation remains plausible, what evidence should be elicited next, and
  how it should change the teaching decision.
- Identify three ranked diagnostic priorities, or explain why fewer are
  defensible. Name concepts or skills rather than question numbers, levels, or
  scripts.
- For each priority, connect: **located observed pattern → curriculum construct
  or prerequisite → diagnostic claim and teaching implication → ranking
  rationale**.
- Label and justify each claim as **supported**, **provisional**, or
  **unresolved**. Explain how the proposed mechanism could produce the pattern,
  rank comparatively by prevalence, impact and/or prerequisite importance, and
  explain why Priority 3 does not displace the top two.

For an objective or question-scored assessment, give the supplied per-question
figure and marking-scheme expectation for every item discussed, and connect
item/marking-point evidence and student work to the curriculum construct and
teaching decision.

For a constructed-response or rubric-assessed assessment, reason from
**descriptor/criterion → observable script feature or excerpt → diagnostic
claim → teaching decision**. Use supplied levels and scripts to distinguish
frequent patterns from those blocking an important descriptor boundary. Do not
infer absent item statistics, error rates, domain subscores, or scores.

## 2. Top-two diagnostic decision cards

For each of the top two priorities, provide:

1. **Complementary evidence:** the applicable pair specified below, with precise
   locators.
2. **Uncertainty and rival:** the strongest counterevidence, conflicting
   pattern, missing comparison, or limitation; what remains unsettled; and one
   nearby rival requiring a materially different response, including what the
   evidence does and does not do to distinguish it.
3. **Decision test:** an actual 3–5 minute prompt, item, annotation, or
   micro-revision derived from a cited response, worked step, excerpt, or
   marking point; a way to collect evidence from every learner; the observable
   pattern expected under the main claim and rival; an inspection/scoring rule;
   and the different teaching decision for each result.
4. **First response:** a precise target, opening scaffold or teacher move,
   source-derived example/prompt/exercise/constraint, evidence-defined learner
   group, response feature to change, and observable success threshold. State
   both next-decision branches; if the threshold is missed, change the scaffold,
   regroup, or revisit the diagnosis.

Do not use a provisional claim to prescribe an undifferentiated whole-class
re-teach.

For an objective or question-scored assessment, pair one aggregate, item, or
marking-point anchor with one distinct actual learner answer or worked step.
The decision test must distinguish the main claim from the rival rather than
repeat the original question.

For a constructed-response or rubric-assessed assessment, where the scripts
permit, contrast one stronger and one weaker short supplied-script excerpt at
the same descriptor boundary. Give both anonymised locators, the separating
feature, and what the contrast does not establish. If no valid pair exists,
mark the claim provisional and state the missing comparison. Apply the decision
test to the weaker excerpt or the same boundary. Include in the first response
one concrete feedback, annotation, or micro-revision move applied to the weaker
excerpt, and identify what would count as transfer in fresh writing.

## 3. Remaining decisions and reporting

- For Priority 3, give its current next step and a specific evidence-grounded
  trigger for promotion into the next teaching cycle.
- Include one evidence-supported issue that was considered but deprioritised:
  give its locator, why it loses on prevalence, impact/dependency, or confidence,
```

```
  and what evidence would promote it. If no fourth issue is defensible, say so.
- Define every learner group from supplied score ranges, levels, linked response
  features, or learner/script IDs, and give each group a specific next step.
- Use a compact, scannable evidence-to-action structure. Let learner evidence
  visibly shape diagnosis and action. State evidence and rationale once, omit
  file inventories or process accounts, and remove anything that does not
  change a diagnosis, decision, action, or check.
```

J.2.2 EXAM-ANALYSIS SLIDE CREATION

Listing 2 gives the slide prompt; Section 2 summarizes its requirements.

Listing 2: Producer prompt for the 30-slide exam-analysis task.

```
# Identity

You are an expert senior-high-school teacher, exam-data analyst, and presentation
     designer. You create clear, accurate, and student-facing exam-analysis slides that
     help teachers explain class performance and help students understand their mistakes
     and improve.

# Instruction

Create a 30-slide PowerPoint deck for a classroom scenario in which a teacher presents
     and explains exam results to senior high school students.

1. Use the skill `ppt-master`.
2. All required data is located in the `source_materials` folder.
3. The entire slide deck must be written in English.
4. You are currently executing this task in an automated pipeline. I agree that you may
     start creating the PowerPoint directly without asking for further confirmation.

The presentation should contain three main parts:

## 1. Overall exam data and performance overview

1. Present and analyze the key statistics of this exam:
   1. Show maximum score, average score rate, highest score rate, lowest score rate, and
     score distribution. Include other relevant metrics where useful.
   2. Analyze class-level performance patterns and identify any notable trends or
     outliers, supporting the observations with relevant statistics.
2. Present and analyze question-level data:
   1. Show maximum score, average score rate, total marks lost by the class,
     discrimination, and attempt rate. Include other relevant indicators where useful.
   2. Show two views of question performance: rank top-level questions by average error
     rate (`1 - average score rate`) and separately by the total marks lost by the class.

Use charts, tables, or visual summaries to present the data and analysis clearly.

## 2. Detailed analysis of the five top-level questions with the highest average error
     rates

Select the five top-level questions with the highest average error rates. For each
     question:

1. Performance analysis:
   1. Analyze sub-question performance and identify some sub-questions that contributed
     substantially to the total marks lost.
   2. Examine actual student answers and show the rates of blank, fully incorrect,
     partially correct, and full-mark responses.
2. Question content analysis:
   1. Explain the accepted answer and solution process.
   2. Analyze the question focus, key knowledge points, and the reasoning path students
     should follow for this type of question.
3. Analyze common errors from two perspectives:
   1. Show common error patterns with counts and short examples.
   2. Identify some scoring points that were missed by a large number of students. For
     each point, show the number of students who missed it and the associated total marks
     lost.
4. Common mistakes analysis:
   1. Analyze the likely causes of common mistakes and provide strategies for avoiding
     similar errors in future exams.
   2. Choose representative student responses and show how to revise them into
     full-credit responses aligned with the rubric.

## 3. Cross-question diagnosis, summary, and suggestions

1. Identify knowledge, reasoning, data-handling, or answer-construction problems that
     recur across multiple selected questions. Support each finding with question numbers,
```

```
     rubric evidence, student counts, and the associated total marks lost. Separate
     high-frequency problems (high student count) from high-impact problems (high total
     marks lost).
2. Identify which pairs of selected questions most often caused difficulty for the same
     students (defining difficulty as a score rate below 50%).
3. Summarize the class's strengths and main areas for improvement in an objective and
     encouraging tone. Support each point with relevant statistics and analysis.
4. Provide a short review plan with clear priorities and success checks. Support each
     item with relevant statistics and analysis.
```

J.2.3 PERSONALIZED STUDENT-WORKSHEET CREATION

Listing 3 gives the student-worksheet prompt; Section 2 summarizes its requirements. It specifies one DOCX/PDF pair per learner, exactly three focus questions with matched practice questions, and a two-page limit. The selection rule it prescribes is formalized below.

Question selection. For student i and question q, let r_{iq} be earned marks divided by available marks, and let C_{iq} contain students who sat the same section and have a recorded score on q, including i. For English Language, the comparison group is defined by the student's B1/B2 route, including for questions in the common section. The prompt defines

$$c_{iq} = \frac{1}{|C_{iq}|} \sum_{j \in C_{iq}} r_{jq}, \qquad g_{iq} = c_{iq}(1 - r_{iq}). \tag{6}$$

Among questions with $r_{iq} < 1$ and $c_{iq} \geq 0.30$, higher g_{iq} values guide the three requested focus choices; agents may choose among ties. When fewer than three questions qualify, or a scored subpart could count as a separate focus, the prompt leaves that judgment to the producing agent. This selection favors a larger student shortfall on an item that the comparison group handled relatively well, without weighting by the item's raw mark total. For example, a class mean of 0.80 and student ratio of 0.25 give $g = 0.60$, ahead of $g = 0.40$ for a zero-scored item with class mean 0.40. Class rate is a mean score ratio, not the proportion of students earning full marks, and a selected item need not be one on which the student scored below the class mean.

Mathematically, this rule can be viewed as a task-specific adaptation of the weighted-error component in student–problem (S–P) caution analysis (Harnisch & Linn, 1980, p. 4). That component weights a binary error by the number of students answering the item correctly. We replace binary answers with score ratios and the correct-answer count with a same-section mean, then rank all eligible items individually. The resulting heuristic prioritizes potential foundational gaps; unlike the full S–P analysis, it does not compute the person-level caution index. The 0.30 cutoff limits attention to items with some cohort success, and three focus items leave room for explanation and matched practice within two pages. Both values are task-design choices rather than tuned thresholds. For each focus item, the agent supplies the question, the student's answer where available, the correct answer or method, an encouraging explanation of the error and its repair, and the tested knowledge or skill. Each receives one matched practice question with an answer key.

Listing 3: Producer prompt for personalized student-worksheet creation.

```
# Identity

You are an expert senior-high-school teacher, exam-data analyst, and document designer.
     You create clear, accurate, and student-facing error sheets that show each student
     their own most important mistakes, explain them in plain language, and give them
     matched practice.

# Instruction

Create one personalised A4 error sheet per student for a classroom scenario in which a
     teacher hands back a marked exam paper to senior high school students.

1. All required data is located in the `source_materials` folder.
2. Author each sheet as a Word document (`.docx`) and export its PDF from that same file
     with LibreOffice headless (`soffice --headless --convert-to pdf`). Deliver both files
     for every student.
   - Write them into the `outputs/` folder of your workspace — the only place
     deliverables are collected from — and put nothing else there.
   - Name each pair after that student's `student_id`, with any space replaced by an
     underscore: `<student_id>.docx` and `<student_id>.pdf`.
```

```
3. Each sheet is at most two A4 pages, for double-sided printing, and must be written in
     English.
4. You are currently executing this task in an automated pipeline. I agree that you may
     start creating the sheets directly without asking for further confirmation.

Each sheet should contain four main parts:

1. Student information
   - The exam's subject, paper and year; the student's anonymous id; and the section or
     route they sat where the paper offers alternatives.

2. Score overview
   - The student's total score, and which questions were full marks, partial, or zero.
   - Present it so it stays readable when printed in black and white.

3. Error focus — exactly the three questions that matter most for this student
   - Choose them by foundational gap rather than by marks lost: rank the student's
     non-full questions by `class_rate × (1 − student_ratio)` — the class's mean score
     ratio on the question against this student's own — excluding questions where
     `class_rate < 0.30`, and compare each student only with those who sat the same
     section.
   - For each question: the question itself, the correct answer or solution method, and
     the student's own answer where it is available.
   - An explanation of why the student's answer is wrong and how to get to the right one,
     in plain and encouraging language a 17-year-old reads easily.
   - The question's domain, type, and the basic knowledge or skill it tests.

4. Practice
   - Exactly three practice questions, one matched to each focus question in the same
     type and topic, each with a worked answer.
   - Print all practice answers upside-down at the bottom of the last page.
```

Further prompts. The evaluator prompts and checklists for the four evaluation dimensions are provided in our repository at `https://anonymous.4open.science/r/Exam2TeachAnonymous-8E75/`.

K Additional Related Work

Educational foundations and applications. Formative assessment and feedback connect evidence of student understanding to instructional decisions and learner improvement (Black & Wiliam, 1998; 2009; Sadler, 1989; Hattie & Timperley, 2007), and learning analytics scales this connection with learner data (Pardo et al., 2019; Kyne et al., 2023). AI systems now assist with diagnostic reports, learner-adapted worksheets, and course materials (Xie et al., 2025; Gonnermann-Müller et al., 2025; Yao et al., 2026), motivating our focus on whether generated materials address the class and individual needs documented in the evidence.

LLM agents and execution frameworks. Agents act through tools, code, and computer interfaces (Yao et al., 2023; Wang et al., 2024a; Yang et al., 2024; Wang et al., 2025a), and sustain long tasks through multi-agent coordination (Wu et al., 2024; Hong et al., 2024; Fourney et al., 2024), memory and context management (Packer et al., 2023; Shinn et al., 2023; Mei et al., 2025; Wang et al., 2025b), and task-state tracking, deliverable checking, and recovery (Ma et al., 2026; Zheng et al., 2026; Zhuang et al., 2026). Outputs are refined with model feedback, tools, retrieval, and verification questions (Madaan et al., 2023; Gou et al., 2024; Gao et al., 2023; Dhuliawala et al., 2024), although self-correction without external feedback can fail to improve reasoning (Huang et al., 2024). Since these mechanisms shape the final materials, Exam2Teach evaluates complete model–agent systems by their outputs, with image, inputs, and tools held fixed across models and harnesses.

L Limitations

Exam2Teach covers one assessment in each of five subjects from a single examination system, and evaluation targets document quality against a teacher-informed bar rather than measured classroom outcomes. Extending to further assessments, curricula, and languages, and calibrating score weights and pass thresholds against teacher revision time and student learning, are natural next steps. Each of the 150 main-experiment runs is generated once per configuration, with up to three attempts, and

the ablations compare single Biology outputs (Appendix B). Because the main experiment alone consumes about 1.4 billion producing-model tokens and roughly US$600 in estimated generation cost, with Claude Opus 5 alone averaging US$21.01 per run (Appendix C), we prioritised breadth over repetition; repeated runs would sharpen finer rankings and subject-level contrasts. Documents are rendered through LibreOffice for evaluation, so layout results may differ slightly from how the same files display in Microsoft Office. Human validation of the evaluators is deepest for Biology and slides, and several judge models also generate benchmark outputs (Appendix E.4); broader review across subjects and formats, and judges disjoint from the generating models, would further tighten reliability estimates.