Instructional Agents: Reducing Teaching Faculty Workload through Multi-Agent Instructional Design

Huaiyuan Yao*, **Wanpeng Xu***, **Justin Turnau, Nadia Kellam, Hua Wei**
Arizona State University
{huaiyuan, wanpeng.xu, jturnau, nadia.kellam, hua.wei}@asu.edu
* Equal Contribution

Abstract

Preparing high-quality instructional materials remains a labor-intensive process that often requires extensive coordination among teaching faculty, instructional designers, and teaching assistants. In this work, we present Instructional Agents, a multi-agent large language model framework designed to automate end-to-end course material generation, including syllabi creation, LaTeX-based slides, lecture scripts, and assessments. Unlike prior tools focused on isolated tasks, Instructional Agents simulates role-based collaboration to ensure pedagogical coherence. The system operates in four modes: Autonomous, Catalog-Guided, Feedback-Guided, and Full Co-Pilot mode, enabling flexible control over the degree of human involvement. We evaluate Instructional Agents across five university-level courses and show that it produces high-quality instructional materials that are reviewed and refined by teaching faculty prior to use, while significantly reducing the time required to prepare classroom-ready content. By supporting institutions with limited instructional design capacity, Instructional Agents provides a scalable and cost-effective framework to democratize access to high-quality education, particularly in underserved or resource-constrained settings. The project website, including source code, is available at https://darl-genai.github.io/instructional_agents_homepage/

1 Introduction

The preparation of instructional materials is a fundamental but labor-intensive aspect of education (Merritt, 2016; Gavin and McGrath-Champ, 2024). Instructors must design syllabi, create slides, and develop teaching notes, which often require coordination among faculty, instructional designers, and teaching assistants. Despite its pedagogical importance, the process is manual and time-consuming, limiting scalability. The absence of instructional design support exacerbates these challenges, often resulting in high preparation costs even for routine course development.

Recent advances in large language models (LLMs) have sparked growing interest in AI-assisted education (Wang et al., 2024; Baig and Yadegaridehkordi, 2024). While AI tools have addressed isolated tasks such as tutoring and grading (Zhai, 2023), they lack end-to-end workflows for instructional design. As a result, instructors still invest substantial effort in producing coherent course materials, often resulting in fragmented alignment between objectives, assessments (e.g., quizzes, exams, and peer-reviewed assignments), and content (Biggs, 1996; Wang et al., 2013; Biggs et al., 2022).

To address these challenges, we introduce Instructional Agents, a multi-agent LLM framework for automated course material generation. Unlike single-model approaches, Instructional Agents simulates collaborative workflows among a comprehensive group of educational roles, including Teaching Faculty, Instructional Designer, Teaching Assistant, Course Coordinator, and Program Chair. These agents interact guided by the instructional design framework, ADDIE (Gagne et al., 2005; Branch and Varank, 2009), ensuring alignment across learning objectives, assessments, and content. Instructional Agents also supports four modes: Autonomous, Catalog-Guided, Feedback-Guided, and Full Co-Pilot. These modes allow for a balance between automation and human involvement. By mimicking real-world instructional collaboration, the system aims to preserve instructional coherence while scaling the design process.

This paper investigates whether multi-agent LLM systems can support instructional material generation in higher education. We evaluate how interaction modes impact output quality, efficiency, and scalability, with a focus on reducing faculty workload while preserving pedagogical rigor.

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4087–4109

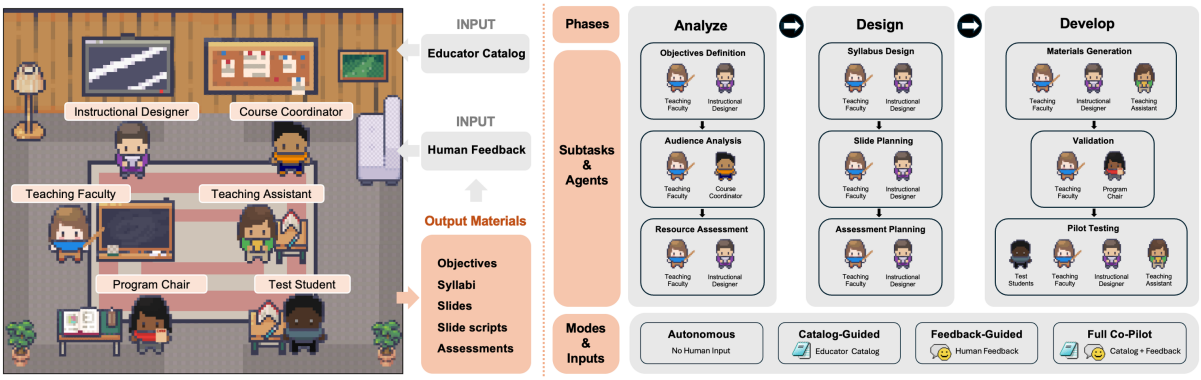


Figure 1: **Overview of Instructional Agents.** (Left) Inputs and outputs in Instructional Agents. Educator input and human feedback guide the generation of key instructional materials, including learning objectives, syllabi, slides, slide scripts, and assessments. (Right) Instructional Agents framework showing the overall workflow based on the first three phases of the ADDIE instructional design framework (Gagne et al., 2005; Branch and Varank, 2009): Analyze, Design, and Develop. Within each phase, multiple role-specialized agents (Teaching Faculty, Instructional Designer, Teaching Assistant, Course Coordinator, and Program Chair) collaborate through structured prompt exchanges to complete subtasks and refine outputs in an iterative workflow. Appendix F provides the specific prompts used for each type of agent. Each prompt includes a tailored background context and clearly defines the agent's goals, tasks, and responsibilities to ensure coherent and role-aligned response generation. Avatars are illustrative and designed for diversity without implying real demographic proportions or stereotypes.

Rather than student outcomes, we assess the revision effort required by teaching faculty. In summary, our contributions are as follows:

- We present Instructional Agents, a multi-agent LLM framework for automated course material generation, structured around the ADDIE instructional design framework.
- We introduce role-based collaboration among educational agents with different levels of human involvement to ensure coherent, pedagogically aligned content. Specifically, Instructional Agents supports comprehensive roles in instructional design and operational modes to flexibly balance automation and human oversight.
- We evaluate Instructional Agents on five courses using both human and automated reviewers, showing that it reduces educator workload and preserves rigor and coherence, while also revealing trade-offs between automation, quality, and scalability.

2 Background and Related Work

LLM Agents and Role Collaboration Large Language Models have enabled the creation of multi-agent systems where model instances assume distinct roles to collaborate on complex tasks (Yao et al., 2025a; Zhao et al., 2025). These systems have been applied to domains such as scientific research (Ma et al., 2024; Schmidgall et al., 2025), web automation (Yang et al., 2024), and interactive behavior simulation (Park et al., 2023; Yao et al., 2025b), demonstrating that structured agent interaction improves task consistency and division of labor (Wang et al., 2025; Rasal and Hauer, 2024). However, most applications optimize for factual correctness or task success, without addressing pedagogical alignment or coherence. The education domain presents distinct challenges, requiring collaboration across instructional roles and consistency among diverse outputs, which are not addressed by typical LLM agent pipelines (Chu et al., 2025).

LLMs in Education In education, recent studies have focused on classroom simulation and task-specific automation. For example, LLM agents have been used to emulate teacher-student dialogues for training and research (Zhang et al., 2024; Hao et al., 2025; Hu et al., 2025). Others target automation of instructional tasks such as syllabus drafting, lesson planning, or content review (Davis and Lee, 2023; Fan et al., 2024; Roodsari and Ghanbari, 2024). While these systems show promise, they often operate in isolation and lack integration into broader instructional pipelines. Current applications often fall short in ensuring educational rigor and alignment with pedagogical goals (Kasneci et al., 2023). Recent work continues to critique AI applications in education for their limited pedagogical grounding and lack of integration with established instructional frameworks (Zawacki-Richter et al., 2024).

Instructional Design and Automation Instructional design frameworks such as ADDIE (Gagne et al., 2005; Branch and Varank, 2009) emphasize structured development across phases: Analyze, Design, Develop, Implement, and Evaluate. These frameworks offer a clear structure, but their real-world adoption is limited. Many instructors strug-

gle to translate such models into practice due to time constraints and lack of support (Bennett et al., 2017). Others point to deeper institutional barriers, including insufficient incentives and tensions with professional identity (Brownell and Tanner, 2012). While recent LLM-based tools have shown potential to automate isolated instructional tasks, they are typically single-pass and operate without reference to instructional design frameworks or cross-role coordination. Our approach embeds the ADDIE structure into a multi-agent LLM framework that simulates collaboration among instructional roles and supports pedagogically aligned material generation across the full course development pipeline.

3 Method: System Design and Workflow

To support collaborative instructional design, we propose Instructional Agents, a multi-agent LLM system that automates course content generation through role-specialized collaboration. The system simulates common educational roles involved in course development, including Teaching Faculty, Instructional Designer, Teaching Assistant, Course Coordinator, and Program Chair. Among these roles, the Teaching Faculty agent serves as the primary authority and maintains continuous oversight throughout the entire workflow, while other agents provide complementary support for structure, implementation, validation, and feedback. Together, these agents operate within a structured workflow inspired by the ADDIE instructional design framework to produce a coherent and instructionally aligned course package, including learning objectives, syllabi, assessments, slide content, and slide scripts.

3.1 Workflow Overview

Figure 1 illustrates the overall end-to-end workflow of the system. To clarify how the workflow operates in practice, we first summarize the full process before describing individual components in detail. The workflow consists of three sequential phases: *Analyze*, *Design*, and *Develop*, corresponding to the first three stages of the ADDIE framework.

In the *Analyze* phase, the Teaching Faculty agent leads the formulation of instructional objectives and instructional intent. The Course Coordinator agent supports this process by providing course-level context, constraints, and background information, such as student characteristics and resource limitations. Together, these activities result in an Instructional Foundation Report, which serves as a shared grounding artifact for all subsequent phases.

In the *Design* phase, the Teaching Faculty agent continues to guide pedagogical decisions, ensuring alignment between objectives, content, and assessments. The Instructional Designer agent supports this phase by structuring the syllabi, organizing instructional flow, and refining assessment plans. The outputs of this phase include structured syllabi, key instructional points, and draft assessments that define the course's pedagogical plan.

In the *Develop* phase, the Teaching Assistant agent generates concrete instructional materials, including slides, slide scripts, and finalized assessments, under the guidance of the Teaching Faculty agent. The Program Chair agent then reviews the generated materials from a broader program-level perspective to provide validation and suggestions, and the Test Student agent supplies simulated learner feedback to support iterative refinement. The final outputs of this phase are refined instructional objectives, syllabi, slides, slide scripts, and assessments, forming a cohesive and instructionally aligned course package.

While the ADDIE framework formally includes additional *Implement* and *Evaluate* phases, this work focuses on the first three phases due to practical and ethical considerations, including the need for human oversight before deploying AI-generated instructional materials to real students.

3.2 Analyze Phase

The Analyze phase focuses on understanding the instructional goals, learner profiles, and logistical constraints. It consists of three subtasks:

Objectives Definition In this subtask, the Teaching Faculty and Instructional Designer collaborate to define competency-aligned course objectives. The Teaching Faculty agent initiates goal proposals based on domain knowledge, while the Instructional Designer ensures alignment with accreditation standards and instructional best practices.

Audience Analysis In the *Audience Analysis* subtask, the Teaching Faculty agent works with the Course Coordinator agent to build a learner profile by analyzing student backgrounds, prior knowledge, and challenges. This helps shape prerequisites and instructional strategies.

Resource Assessment In this subtask, the Teaching Faculty agent assesses teaching needs, and the

Instructional Designer evaluates institutional constraints (e.g., platform compatibility). Together, they define feasible instructional strategies.

3.3 Design Phase

The Design phase organizes the course structure and assessment strategy. Agents collaborate to create syllabi with weekly topics, outline instructional methods, and align assessments with learning objectives. Feedback to support formative evaluation is also planned in this phase.

Syllabus Design As shown in Figure 1, in the *Syllabus Design* subtask, the Teaching Faculty and Instructional Designer agents jointly develop course syllabi, including weekly topics, readings, and assignments. This subtask uses the previously defined objectives and learner profile to structure the course timeline. The output is syllabi that specify content coverage, assessment milestones, and delivery modes, which can be used for subsequent content development.

Slide Planning In this subtask, the Teaching Faculty and Instructional Designer agents co-develop the instructional flow for each weekly topic. The process begins with identifying key concepts and logical sequences based on the previously defined objectives and learner profile. The Teaching Faculty agent drafts initial slide content, including conceptual explanations, technical examples, and transitional narratives. The Instructional Designer agent then refines this content by structuring it for clarity, pedagogical flow, and visual coherence. The result is a slide content plan that serves as the foundation for material development in the subsequent phase. This process is visually summarized in the instructional workflow diagram shown in Figure 2, which bridges the planning and generation stages across the Design and Develop phases.

Assessment Planning During the *Assessment Planning* stage, the Teaching Faculty and Instructional Designer agents collaboratively define assessment strategies that align with course objectives. They design a multi-stage capstone project to replace traditional summative exams, incorporating deliverables such as a proposal, progress report, and final submission. Additionally, they establish formative feedback mechanisms, including peer review checkpoints, integrity guidelines, and grading rubrics. These assessments are integrated into the course timeline to ensure alignment with

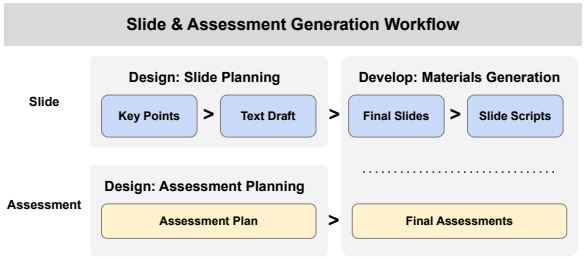


Figure 2: Workflow of slide and assessment generation from key points and drafts to final slides, slide scripts, and assessments across the Design and Develop phases.

instructional goals and provide ongoing support for student learning. The output informs material generation and provides guidance for classroom use by Teaching Faculty.

3.4 Develop Phase

The Develop phase transforms the instructional plans from the Design phase into complete, classroom-ready materials. To ensure that the instructional outputs are pedagogically aligned, accurate, and usable, as shown in Figure 1, we design three interconnected stages:

Materials Generation In this subtask, the Teaching Faculty, Instructional Designer, and Teaching Assistant collaborate to generate all core instructional materials. For each chapter derived from the syllabi, the agents produce final LaTeX-based slides, slide scripts, and assessments.

The process begins by transforming slide planning outputs (i.e., key points and text drafts) into final slides and scripts. The Teaching Faculty agent expands the content with technical explanations and examples, the Instructional Designer structures the materials for pedagogical flow, and the Teaching Assistant formats them into LaTeX documents. Similarly, assessment plans from the Design phase are converted into final assessments, including quizzes, milestones, and grading rubrics.

Once all content materials are finalized, the system performs a LaTeX compilation step to render the materials into publishable PDF packages. This integration process is handled by a dedicated *LaTeX Compiler* module, which ensures consistent layout and formatting.

Validation The validation subtask involves expert review by the Teaching Faculty and Program Chair agents. All generated materials, including slides, slide scripts, and assessments, are reviewed for pedagogical alignment, content accuracy, and compliance with institutional expectations. The

Program Chair agent provides suggestions and approval notes, which are incorporated by the Teaching Faculty or Instructional Designer agents before finalization. This ensures the materials meet program-level quality standards. This step models how real instructors would revise and approve materials before using them in the classroom. No generated material is assumed ready for deployment without human oversight.

Pilot Testing To further evaluate usability, the system performs a pilot testing stage involving simulated student agents. These test agents engage with instructional materials under controlled scenarios. The Teaching Faculty, Instructional Designer, and Teaching Assistant agents monitor the interactions and identify issues such as confusing phrasing, misaligned pacing, or missing prerequisite knowledge. Feedback collected during this stage informs final refinements before deployment.

3.5 Modes of Operation

Instructional Agents supports multiple modes of operation, each designed to accommodate different levels of human involvement and prior knowledge integration. The system can operate in four different modes, each with a different level of human input: Autonomous Mode (Auto), Catalog-Guided Mode (Cat), Feedback-Guided Mode (Feed), and Full Co-Pilot Mode (Pilot). These abbreviations are used in figures, tables, and other space-limited contexts for clarity.

Autonomous Mode In this mode, the system proceeds through all deliberations and content generation steps without human intervention beyond the initial course name or topic input. Each agent executes its role, moving from one subtask to the next upon completion. The agents autonomously generate learning objectives, syllabi, assessments, slides, and slide scripts. This mode is fully automated and suitable for baseline benchmarking or rapid prototyping of course content.

Catalog-Guided Mode Under this mode, the system incorporates pre-existing institutional or instructor-provided data as `Educator_Catalog` to guide the deliberations. For example, predefined course structures, institutional policies, prior student feedback, and teaching constraints can be included in the `Educator_Catalog` and passed to agents during initialization. These inputs inform agents' decisions, enabling the system to align outputs with existing curricula or departmental guidelines. This mode ensures continuity with institutional practices and reduces the risk of generating content that conflicts with prior standards. A sample catalog is provided in Appendix C.

Feedback-Guided Mode This mode aims to enable retrospective correction and refinement of generated outputs. After a deliberation is completed, a human reviewer can inspect the results and provide targeted suggestions for improvement. The system supports rerunning individual deliberations with the new suggestions appended to the original context. This mode allows for iterative revision of specific materials, such as modifying assessment plans, without restarting the entire pipeline.

Full Co-Pilot Mode To simulate a collaborative workflow between the human teaching faculty and the agent system, in Full Co-Pilot Mode, the system pauses at the end of each subtask to solicit user feedback before proceeding. The user can approve the current outputs, request modifications, or provide guidance for the next steps. In addition to real-time feedback, this mode also incorporates structured preferences through the same `Educator_Catalog` used in Catalog-Guided Mode. These catalog entries allow the system to maintain alignment with institutional policies and instructor intent across multiple subtasks, such as emphasizing specific topics in the syllabi, adjusting slide content focus, or altering assessment styles. By combining catalog initialization and human-in-the-loop feedback, Full Co-Pilot Mode closely mirrors real-world curriculum development, where iterative human review and prior knowledge are both integral to quality assurance.

3.5.1 Summary of Modes

These modes provide flexible control over the instructional design pipeline, ranging from fully autonomous execution to human-in-the-loop collaboration. Importantly, in all human-in-the-loop modes, Teaching Faculty retain control over final approval, ensuring that AI-generated content serves as a draft for human refinement. By enabling initialization from prior teaching artifacts, post-generation feedback integration, and interactive human collaboration, Instructional Agents supports a wide range of content development scenarios across instructional contexts.

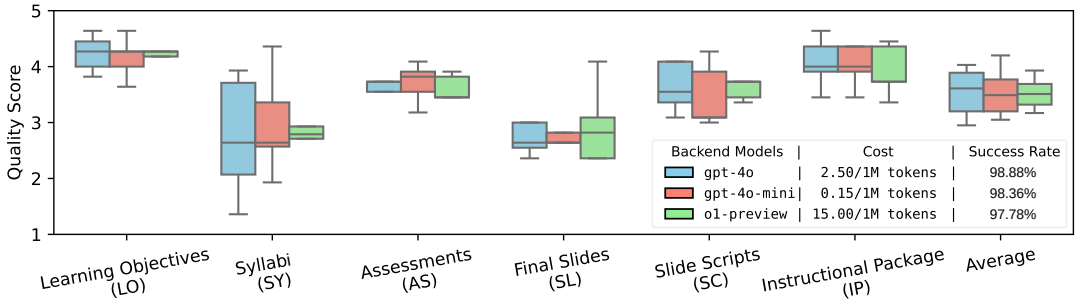


Figure 3: **(RQ1) Quality evaluation of generated instructional materials across different model backends with their costs and success rates.** This table reports the adapted QM-based rubric scores for course materials generated by Instructional Agents using three LLM backends: gpt-4o, gpt-4o-mini, and o1-preview. The evaluation is on six instructional outputs Instructional Agents generated: Learning Objectives (LO), Syllabi (SY), Assessments (AS), Final Slides (SL), Slide Scripts (SC), and the overall Instructional Package (IP). Scores are averaged over five human evaluators for each of the five courses. Each cell represents a score on a 1–5 Likert scale, where higher is better. gpt-4o-mini achieves a performance level and success rate comparable to gpt-4o and o1-preview, while offering the lowest cost. Detailed numbers are provided in Appendix D.1.

4 Experiments

In this section, we present our experimental evaluation around three research questions:

- **RQ1:** How should we evaluate the quality of AI-generated instructional materials? How do human evaluations compare to LLM-based automated assessments?
- **RQ2:** How do different operational modes (Autonomous, Catalog-Guided, Feedback-Guided, Full Co-Pilot) affect instructional quality and instructor workload?
- **RQ3:** What are the runtime and cost trade-offs across different operational modes?

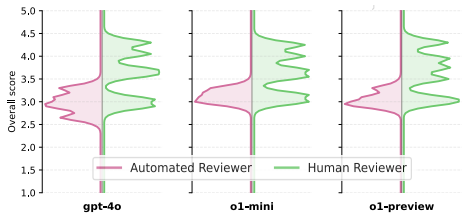
In this section, we additionally report ablations on different agent roles as well as success rate across different procedures. Further experimental results, including the influence of additional backend model evaluations (e.g., LLaMA, Qwen) and an ethics evaluation, are provided in Appendix D.

4.1 Experimental Settings

Model Backends We test the following model backends for content generation: gpt-4o (OpenAI, 2024a), gpt-4o-mini (OpenAI, 2024b), o1-preview (OpenAI, 2024c). To evaluate the framework, we apply Instructional Agents to five university-level courses that vary in structure and depth. The courses include Data Mining, Foundations of Machine Learning, Data Processing at Scale, Introduction to Artificial Intelligence, and Topics in Reinforcement Learning. Detailed hyperparameters are reported in Appendix A. We also test open-source models and report their results in Appendix D.2 and do not find that open-source models show superior performance. Therefore, in later parts of this paper, we primarily test using the above three GPT models.

Evaluation Criteria We adapt the Quality Matters (QM) Higher Education Rubric, Seventh Edition (Quality Matters, 2023), a widely used framework for quality assurance in online and hybrid course design, to evaluate instructional materials at the component level. While the original QM Rubric emphasizes holistic course structure, our version is guided by domain experts in instructional design and higher education and customizes selected QM dimensions to assess six key outputs generated by Instructional Agents: Learning Objectives (LO), Syllabi (SY), Assessments (AS), Final Slides (SL), Slide Scripts (SC), and the overall Instructional Package (IP). Each output is evaluated using a set of tailored metrics, such as clarity, alignment, and variety, which are designed to reflect its specific pedagogical role. Human evaluators rate each item on a 5-point Likert scale based on the revision effort required. The full scoring criteria, along with the mapping from Quality Matters (QM) dimensions to our adapted evaluation metrics, are provided in Appendix B.

Evaluator We apply two kinds of evaluators: (1) *Human Reviewer*. For each course, we recruit five expert instructors to serve as human evaluators. These include faculty members and senior PhD students with prior teaching experience. Each evaluator rates the instructional package using six adapted criteria described above. (2) *Automated Reviewer*. In addition to human evaluation, we employ different LLMs (gpt-4o, gpt-4o-mini, and o1-preview) as automated reviewers to evaluate the generated materials using a rubric-based prompt aligned with the adapted QM-inspired metrics.



(a) Comparison of evaluation score distribution

Model	Learning Objectives (LO)	Syllabus (SY)	Assessments (AS)	Final Slides (SL)	Slide Scripts (SC)	Instructional Package (IP)
gpt-4o	2.63 (±0.31)	3.12 (±0.17)	2.97 (±0.03)	2.90 (±0.06)	3.28 (±0.06)	3.04 (±0.03)
gpt-4o-mini	3.10 (±0.25)	3.20 (±0.14)	2.99 (±0.05)	2.99 (±0.04)	3.30 (±0.08)	3.09 (±0.03)
o1-preview	2.96 (±0.27)	3.10 (±0.11)	2.96 (±0.04)	2.97 (±0.08)	3.29 (±0.06)	3.06 (±0.05)

(b) Automated LLM reviewer scores

Model	Learning Objectives (LO)	Syllabus (SY)	Assessments (AS)	Final Slides (SL)	Slide Scripts (SC)	Instructional Package (IP)
gpt-4o	4.24(±1.16)	2.74(±1.27)	3.60(±1.30)	2.93(±1.25)	3.64(±1.27)	4.07(±0.81)
gpt-4o-mini	4.16(±1.22)	2.97(±1.36)	3.71(±1.34)	2.91(±1.20)	3.47(±1.29)	4.02(±0.75)
o1-preview	4.24(±1.13)	2.86(±1.40)	3.49(±1.14)	2.95(±1.21)	3.69(±1.16)	3.93(±0.76)

(c) Human reviewer scores

Figure 4: Comparison of evaluation scores (Human reviewer vs. Automated reviewer). (a) The distribution of scores generated by human reviewer and automated reviewer. (b) The scores of LLMs evaluating their own generated instructional materials. Each cell shows the mean (standard deviation) over five courses. Scores are on a 1–5 scale, where higher is better. (c) The scores of human reviewers evaluating instructional materials generated different LLMs. Human reviewers tend to have more diverse evaluations while automated reviewers tend to give mediocre scores.

4.2 Evaluating Instructional Quality: Human vs. LLM Reviewers (RQ1)

We begin by addressing foundational questions about how to evaluate instructional content quality (RQ1), since this choice affects all subsequent experiments on comparisons of models and operational modes. We analyze how different LLM backends influence the quality of generated materials and examine the alignment between automated LLM evaluations and human assessments. We report the following observations:

• **Influence of Backend Model:** Figure 3 presents the evaluation results for six instructional materials and shows that all three backends produce high-quality content, with gpt-4o-mini matching the performance of gpt-4o and o1-preview while offering the lowest cost. A Friedman test confirmed no significant differences among the models (test statistic $Q = 0.473$, p-value $= 0.789$), supporting our conclusion (Friedman, 1937). Given this and the substantially lower computational cost and inference time of gpt-4o-mini, we use it as the default backend model in the remainder of this paper. Detailed statistics are provided in Appendix D.1.

• **Human vs. LLM Evaluations:** Figure 4 compares the distribution of overall evaluation scores assigned by automated LLM reviewers and human evaluators. LLMs consistently assign moderate and tightly clustered scores (typically between 2.9 and 3.1) while human evaluators produce a broader range of scores, demonstrating greater sensitivity to instructional effectiveness. This discrepancy highlights the limited capacity of LLM-based evaluators in distinguishing between higher and lower quality outputs (Hong et al., 2026). Therefore, in the remainder of our experiments, we rely on human assessments as the primary reference for instructional quality. We also conducted the automated reviews across different LLMs, and the results are consistent with Figure 4. Their detailed results can be found in Appendix D.3.

4.3 Impact of Operational Modes (RQ2)

We examine how different operational modes in Instructional Agents influence instructional quality and human workload. The four operational modes differ in how they balance automation with human oversight. For each of the five courses, we generate instructional materials under all four modes and collect human ratings based on the adapted QM rubric. In addition to quantitative scores, we also collect open-ended qualitative feedback from evaluators, summarized in Appendix D.7. The results are shown in Table 1, Figure 5, and Figure 6 . We present our key findings as follows:

• **Overall Comparison Across Modes:** In Table 1 and Figure 5, Full Co-Pilot Mode consistently achieves the highest quality, improving scores by *0.5 to 0.9 points* over Autonomous Mode, especially in Learning Objectives (LO), Slide Scripts (SC), and overall Instructional Packages (IP). Feedback-Guided Mode strikes a good balance between quality and efficiency, with stronger performance on content-rich components like Assessments (AS) and Slides (SL). In contrast, Catalog-Guided Mode outperforms Feedback-Guided Mode in components related to structure and administrative clarity, including Learning Objectives (LO) and syllabi (SY). This can be attributed to the use of pre-loaded templates and institutional guidelines, which support consistency but may limit depth and adaptability. These results highlight that human involvement improves quality, and each mode offers trade-offs between refinement and effort.

• **Material-level Trends:** As shown in Figure 6, all materials achieve average scores above 3.0, indicating generally acceptable quality across modes. Learning Objectives (LO) and Slides (SL) receive the highest ratings on average, while Slide Scripts (SC) tend to score slightly lower. Notably, Slides (SL) also show lower variance across modes, sug-

Table 1: **(RQ2) Human evaluation on instructional materials across operational modes.** This table reports human ratings for course materials generated by Instructional Agents under four operational modes: Autonomous Mode (Auto), Catalog-Guided Mode (Cat), Feedback-Guided Mode (Feed), and Full Co-Pilot Mode (Pilot). Six key outputs generated by Instructional Agents are evaluated: Learning Objectives (LO), syllabi (SY), Assessments (AS), Final Slides (SL), Slide Scripts (SC), and the overall Instructional Package (IP). Each cell presents the mean rating averaged over five expert instructors per course. Scores are on a 1–5 scale, where the higher the better (Ratings reflect estimated instructor revision effort before classroom deployment). With greater human involvement, the material quality is better, and Full Co-Pilot mode consistently achieves the best performance.

	Course 1				Course 2				Course 3				Course 4				Course 5			
	Auto	Cat	Feed	Co-Pilot	Auto	Cat	Feed	Co-Pilot	Auto	Cat	Feed	Co-Pilot	Auto	Cat	Feed	Co-Pilot	Auto	Cat	Feed	Co-Pilot
LO	3.73	4.13	3.87	**4.40**	3.87	4.07	4.20	**4.27**	3.42	**4.17**	3.75	**4.17**	3.17	**3.75**	3.33	**3.75**	3.58	3.92	3.75	**4.08**
SY	3.10	3.65	3.40	**4.05**	2.90	3.85	3.60	**4.05**	2.81	3.44	3.25	**3.62**	2.94	3.38	2.75	**3.75**	2.94	3.19	**3.56**	**3.56**
AS	3.10	3.45	3.55	**3.70**	2.95	3.45	3.45	**3.70**	2.81	3.31	3.31	**3.38**	2.38	**3.31**	2.88	**3.31**	2.31	3.12	3.00	**3.31**
SL	2.87	3.20	3.27	**3.80**	3.00	3.13	3.40	**3.67**	3.00	3.17	3.25	**3.42**	2.58	**3.42**	3.25	3.33	2.50	3.08	**3.42**	3.33
SC	3.20	3.47	3.67	**3.80**	3.67	3.73	**4.13**	4.07	3.17	3.58	3.42	**3.83**	3.08	3.25	3.25	**3.33**	3.25	3.25	3.42	**3.50**
IP	3.33	3.87	3.87	**4.13**	3.07	3.40	**3.87**	3.80	2.75	**3.50**	3.08	**3.50**	2.25	3.58	3.17	**3.83**	2.50	3.58	3.50	**3.67**
Avg	3.22	3.63	3.60	**3.98**	3.24	3.61	3.78	**3.93**	2.99	3.53	3.34	**3.65**	2.73	3.45	3.10	**3.55**	2.85	3.36	3.44	**3.58**

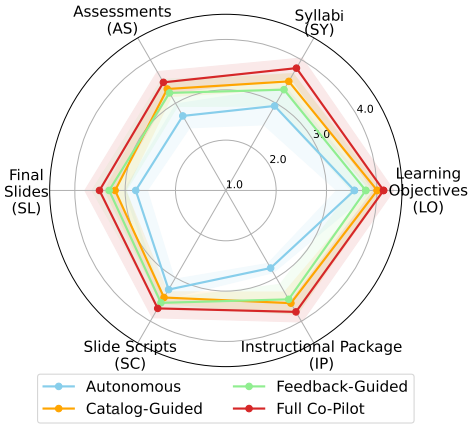


Figure 5: Radar chart analysis on the performance of generating materials at different modes. Each axis represents scores evaluated by human reviewers on one kind of material. Full Co-Pilot mode consistently performs the best.

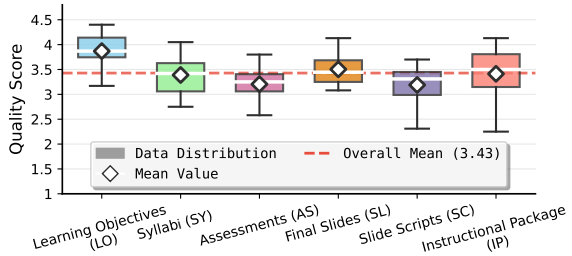


Figure 6: Rating distribution over instructional material types.

gesting that they are more robust to changes in workflow configuration.

4.4 Runtime and Cost Analysis (RQ3)

We evaluate the runtime efficiency and computational cost of Instructional Agents under different operational modes. The evaluation tracks four key metrics: Token Usage, Inference Time, Human Time, and Compute Cost. Human Time reflects instructor involvement for review or co-pilot interaction. All values are averaged across five courses using the gpt-4o-mini backend. Table 2 summarizes the results. Autonomous Mode is the most efficient with lower-quality outputs. Catalog-Guided and Feedback-Guided Modes increase token usage slightly while requiring 10–30 minutes of teaching faculty effort. Full Co-Pilot Mode achieves the highest quality but requires the highest computational and human cost. These results highlight trade-offs between automation and quality: human-in-the-loop modes offer better instructional design outputs at the expense of time and effort.

Table 2: **(RQ3) Runtime and cost analysis across operational modes.** This table presents the runtime, token usage, human effort, and estimated compute cost for Instructional Agents using gpt-4o-mini across four operational modes. The values are averaged over five courses. Inference time and token usage reflect resource consumption, while human time reflects instructor involvement for review or co-pilot interaction. Lower values are better for time, token usage, and cost.

Metric	Auto	Cat	Feedback	Co-Pilot
Token Usage (millions)	1.46	2.05	1.93	2.42
Inference Time (hrs)	2.23	3.73	2.51	4.73
Human Time (mins)	0	10-15	20-30	30-45
Compute Cost (USD)	0.22	0.31	0.29	0.36

4.5 Ablation Studies on Different Agents

To assess the contribution of each agent to the overall instructional design pipeline, we conduct detailed ablation studies by systematically removing individual roles from the multi-agent framework. As shown in Table 3, the single-agent baseline performs the worst overall (Avg = 2.33), highlighting the benefit of role specialization. Removing the Teaching Faculty notably decreases syllabi (SY) and slide (SL) quality, since this agent provides domain knowledge and contextual grounding during the early stages of Analyze and Design. Eliminating the Teaching Assistant results in a moderate quality drop in structural components such as slides and scripts, reflecting this agent's critical role in formatting and LaTeX consistency. The absence of the Instructional Designer causes a sharp decline in learning objectives (LO) and syllabi (SY) clarity, as this agent ensures pedagogical alignment and instructional structure across materials.

Together, these results demonstrate that role specialization is not merely an implementation choice, but a necessary design component for maintaining instructional coherence and quality across artifacts.

Table 3: **Ablation study on the role of agents.**

Method	LO	SY	AS	SL	SC	IP	Avg
Single Agent (GPT-4o-mini)	3.48	2.44	2.06	1.23	2.54	2.24	2.33
w/o Teaching Faculty	3.53	2.12	2.83	1.67	3.83	2.50	2.75
w/o Teaching Assistant	3.57	2.63	2.67	1.83	2.87	3.86	2.91
w/o Instructional Designer	3.02	2.14	2.75	2.85	2.85	3.18	2.80
Ours (Full, Auto)	3.55	2.93	2.71	2.79	3.27	2.78	3.01
Ours (Full, Co-Pilot)	**4.13**	**3.80**	**3.48**	**3.51**	**3.71**	**3.79**	**3.74**

4.6 Success Rate Analysis

Detailed numbers on the distribution of success rates across different procedure can be found in Table 4. Overall, gpt-4o demonstrates better reliability compared to gpt-4o-mini and o1-preview.

Model	Learning Objectives (LO)	Syllabi (SY)	Final Slides (SL)	Slide Scripts (SC)	Assessments (AS)	Avg
gpt-4o	100%	100%	94.4%	100%	100%	98.88%
gpt-4o-mini	100%	100%	91.8%	100%	100%	98.36%
o1-preview	100%	100%	88.9%	100%	100%	97.78%

Table 4: Success rates (%) of different models across various instructional design stages. Failures primarily stem from the generation of a small number of invalid or overly complex LaTeX codes, which lead to `pdflatex` compilation errors (although compilation often succeeds in Overleaf).

5 Discussion

In this section, we reflect the implications of Instructional Agents. For extended analysis, see Appendix E. Instructional Agents demonstrates that high-quality instructional materials can be generated with minimal human input and further enhanced through human-in-the-loop modes. Full Co-Pilot yields the best quality but requires more time and cost, while gpt-4o-mini is the most efficient backend. These results support the system's scalability in resource-constrained settings.

Ethical considerations are central to deployment. While LLMs may introduce bias, human review in Feedback-Guided and Full Co-Pilot modes helps ensure pedagogical soundness. Instructional Agents is designed to support, not replace, Teaching Faculty. Future work should address accessibility, originality verification, and inclusive content design. While it streamlines drafting, final content decisions rely on faculty judgment. Our evaluation quantifies the time Teaching Faculty are likely to save per component.

6 Conclusion

This paper presents Instructional Agents, a multi-agent LLM framework for automating the generation of instructional materials—syllabi, slides, slide scripts, and assessments—through simulated collaboration among educational roles. Evaluations across five courses show that while autonomous workflows reduce time and cost, incorporating human input, especially in Full Co-Pilot mode, improves quality and usability. Catalog-Guided and Feedback-Guided modes offer additional benefits in structural consistency and content depth.

Instructional Agents lowers the barrier to producing materials and enables scalable curriculum development in resource-constrained institutions. Reducing reliance on specialized support promotes broader access to instructional design. This is particularly impactful for community colleges, international programs, and underserved populations, where instructional capacity is limited. Through this work, we aim to support more inclusive, equitable, and globally accessible education systems.

7 Limitations

While our work demonstrates the potential of multi-agent LLM systems for automating instructional material generation, several limitations remain. First, the current framework primarily focuses on the *Analyze*, *Design*, and *Develop* phases of the ADDIE model, without fully addressing *Implementation* and *Evaluation*, which require real-world classroom deployment and longitudinal assessment. Second, the current system has limited support for rich visual and interactive elements, which are important for modern pedagogy. Third, we do not treat bias analysis as a primary evaluation objective in this work. Although all generated materials are subject to faculty oversight, we only conduct an auxiliary bias evaluation using CEAT in Appendix D.5. Finally, the current system incorporates human feedback primarily through regeneration, rather than enabling fine-grained, targeted editing of specific content, which we leave for future work.

8 Acknowledgments

The work was partially supported by NSF award #2442477. We thank Amazon Research Awards, Cisco Research Awards, Google, and OpenAI for providing us with API credits. The authors acknowledge Research Computing at Arizona State University for providing computing resources. The views and conclusions in this paper should not be interpreted as representing any funding agencies.

References

Maria Ijaz Baig and Elaheh Yadegaridehkordi. 2024. Chatgpt in the higher education: A systematic literature review and research challenges. *International Journal of Educational Research*, 127:102411.

Sue Bennett, Shirley Agostinho, and Lori Lockyer. 2017. The process of designing for learning: Understanding university teachers' design work. *Educational Technology Research and Development*, 65:125–145.

John Biggs. 1996. Enhancing teaching through constructive alignment. *Higher education*, 32(3):347–364.

John Biggs, Catherine Tang, and Gregor Kennedy. 2022. *Teaching for quality learning at university 5e*. McGraw-hill education (UK).

Robert Maribe Branch and İlhan Varank. 2009. *Instructional design: The ADDIE approach*, volume 722. Springer.

Sara E Brownell and Kimberly D Tanner. 2012. Barriers to faculty pedagogical change: Lack of training, time, incentives, and... tensions with professional identity? *CBE—Life Sciences Education*, 11(4):339–346.

Zhendong Chu, Shen Wang, Jian Xie, Tinghui Zhu, Yibo Yan, Jinheng Ye, Aoxiao Zhong, Xuming Hu, Jing Liang, Philip S Yu, et al. 2025. Llm agents for education: Advances and applications. *arXiv preprint arXiv:2503.11733*.

Robert O Davis and Yong Jik Lee. 2023. Prompt: Chatgpt, create my course, please! *Education Sciences*, 14(1):24.

Haoxiang Fan, Guanzheng Chen, Xingbo Wang, and Zhenhui Peng. 2024. Lessonplanner: Assisting novice teachers to prepare pedagogy-driven lesson plans with large language models. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–20.

Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.

Robert M Gagne, Walter W Wager, Katharine C Golas, John M Keller, and James D Russell. 2005. Principles of instructional design.

Mihajla Gavin and Susan McGrath-Champ. 2024. Teacher workload and the organisation of work: a research agenda for a post-pandemic future. *Labour and Industry*, 34(1):88–99.

Zhanxin Hao, Fei Qin, Jianxiao Jiang, Jie Cao, Jifan Yu, Zhiyuan Liu, and Yu Zhang. 2025. Ai as learning partners: Students' interactions and perceptions in a simulated classroom with multiple llm-powered agents. In *Proceedings of the 19th International Conference of the Learning Sciences-ICLS 2025, pp. 1789-1793*. International Society of the Learning Sciences.

Yihan Hong, Huaiyuan Yao, Bolin Shen, Wanpeng Xu, Hua Wei, and Yushun Dong. 2026. Rulers: Locked rubrics and evidence-anchored scoring for robust llm evaluation. *arXiv preprint arXiv:2601.08654*.

Bihao Hu, Jiayi Zhu, Yiying Pei, and Xiaoqing Gu. 2025. Exploring the potential of llm to enhance teaching plans through teaching simulation. *npj Science of Learning*, 10(1):7.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. 2023. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274.

Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. 2024. Llm and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. *arXiv preprint arXiv:2405.09783*.

Eileen G Merritt. 2016. Time for teacher learning, planning critical for school reform. *Phi delta kappan*, 98(4):31–36.

OpenAI. 2024a. Gpt-4o model card. https://platform.openai.com/docs/models/gpt-4o.

OpenAI. 2024b. Introducing gpt-4o-mini. https://platform.openai.com/docs/models/gpt-4o-mini. Accessed: 2025-08-01.

OpenAI. 2024c. Introducing o1-preview. https://platform.openai.com/docs/models/o1-preview. Accessed: 2025-08-01.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Jingyang Peng, Wenyuan Shen, Jiarui Rao, and Jionghao Lin. 2025. Automated bias assessment in ai-generated educational content using ceat framework. *Preprint*, arXiv:2505.12718.

Quality Matters. 2023. *Higher Education Rubric, Seventh Edition*. MarylandOnline, Inc. Available at https://www.qualitymatters.org/sites/default/files/PDFs/StandardsfromtheQMHigherEducationRubric.pdf.

Sumedh Rasal and EJ Hauer. 2024. Navigating complexity: Orchestrated problem solving with multi-agent llms. *arXiv preprint arXiv:2402.16713*.

Sam Toorchi Roodsari and Shahram Azizi Ghanbari. 2024. Instructional design and ai in learning environments: Developing competency-validated adaptive feedback for higher education. In *London International Conference On Education*.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Zicheng Liu, and Emad Barsoum. 2025. Agent laboratory: Using llm agents as research assistants. https://arxiv.org/abs/2501.04227.

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

Xiaoyan Wang, Yelin Su, Stephen Cheung, Eva Wong, and Theresa Kwong. 2013. An exploration of biggs' constructive alignment in course design and its impact on students' learning approaches. *Assessment & Evaluation in Higher Education*, 38(4):477–491.

Yaoxiang Wang, Zhiyong Wu, Junfeng Yao, and Jinsong Su. 2025. Tdag: A multi-agent framework based on dynamic task decomposition and agent generation. *Neural Networks*, page 107200.

Ke Yang, Yao Liu, Sapana Chaudhary, Rasool Fakoor, Pratik Chaudhari, George Karypis, and Huzefa Rangwala. 2024. Agentoccam: A simple yet strong baseline for llm-based web agents. *arXiv preprint arXiv:2410.13825*.

Huaiyuan Yao, Longchao Da, Vishnu Nandam, Justin Turnau, Zhiwei Liu, Linsey Pang, and Hua Wei. 2025a. Comal: Collaborative multi-agent large language models for mixed-autonomy traffic. In *Proceedings of the 2025 SIAM International Conference on Data Mining (SDM)*, pages 409–418. SIAM.

Huaiyuan Yao, Pengfei Li, Bu Jin, Yupeng Zheng, An Liu, Lisen Mu, Qing Su, Qian Zhang, Yilun Chen, and Peng Li. 2025b. Lilodriver: A lifelong learning framework for closed-loop motion planning in long-tail autonomous driving scenarios. *Preprint*, arXiv:2505.17209.

Olaf Zawacki-Richter, John YH Bai, Kyungmee Lee, Patricia J Slagter van Tryon, and Paul Prinsloo. 2024. New advances in artificial intelligence applications in higher education? *International Journal of Educational Technology in Higher Education*, 21(1):32.

Xiaoming Zhai. 2023. Chatgpt for next generation science learning. *XRDS: Crossroads, The ACM Magazine for Students*, 29(3):42–46.

Zheyuan Zhang, Daniel Zhang-Li, Jifan Yu, Linlu Gong, Jinchang Zhou, Zhanxin Hao, Jianxiao Jiang, Jie Cao, Huiqin Liu, Zhiyuan Liu, et al. 2024. Simulating classroom education with llm-empowered agents. *arXiv preprint arXiv:2406.19226*.

Lina Zhao, Jiaxing Bai, Zihao Bian, Qingyue Chen, Yafang Li, Guangbo Li, Min He, Huaiyuan Yao, and Zongjiu Zhang. 2025. Autonomous multi-modal llm agents for treatment planning in focused ultrasound ablation surgery. *Preprint*, arXiv:2505.21418.

A Hyperparameters

Table A.1 summarizes the key hyperparameters adopted in our system of Instructional Agents. The hyperparameters are grouped into two categories: (i) *Foundation Model*, which controls the generative behavior of the underlying large language model (e.g., sampling temperature and token penalties), and (ii) *Application*, which specifies task-level settings such as the number of deliberation rounds and the default length of generated slides. These parameters were chosen empirically based on preliminary experiments to balance generation diversity, reliability, and efficiency in instructional material creation.

Table A.1: Hyperparameters for Instructional Agents

Category	Hyperparameter	Value
Foundation Model	Temperature	1.0
	Top-p	1.0
	Presence penalty	0.0
	Frequency penalty	0.0
Application	Deliberation rounds	1
	Default slides length	30

B Evaluation Criteria

We adapt the Quality Matters (QM) Higher Education Rubric (Seventh Edition) to evaluate the quality and usability of instructional materials generated by Instructional Agents. Rather than adopting the rubric for formal accreditation, we extract selected QM dimensions and map them to output-specific evaluation criteria tailored to our system. Our evaluation focuses on the revision effort required to bring each output to a high-quality state.

Each instructional output is assessed by human raters using a 5-point Likert scale:

- **5** – Minimal edits required; ready to use.
- **4** – Minor revisions needed; content is mostly solid.
- **3** – Moderate revisions needed in structure or clarity.
- **2** – Major restructuring or rewriting required.
- **1** – Complete redevelopment needed; not usable as-is.

Table B.1 shows the mapping between each evaluation metric and the original QM standard categories. The rubric is customized for each output type to reflect its instructional function and expected pedagogical alignment.

The evaluation dimensions were designed in collaboration with domain experts in instructional design and higher education. The expert helped select and adapt relevant elements from the official QM rubric, ensuring that each metric is pedagogically grounded and practical for evaluating generated content. For each output type (e.g., syllabi, assessments), we identified key quality indicators that are both observable and actionable for human reviewers. The resulting rubric aims to balance instructional rigor with feasibility in large-scale human evaluation.

All scores reflect the estimated amount of instructor editing required before the materials are ready to teach.

C Educator Catalog Sample

This educator catalog table organizes the instructional context into seven main categories: student profile, instructor preferences, course structure, assessment design, teaching constraints, institutional requirements, and prior feedback. Each field is mapped to representative content from a hypothetical instructor catalog. Details for each field and example values are provided in Table C.1. The table facilitates structured instructional design and can be used as a template for future course profiling and alignment with teaching goals.

D Additional Results

D.1 Influence of Backend Model

Table D.1 (Quantitative Backend Comparison) shows the average quality scores of instructional outputs generated by Instructional Agents using three backend models: gpt-4o, gpt-4o-mini, and o1-preview. Across all six output types, gpt-4o and gpt-4o-mini achieved comparable average scores (both 3.54), with o1-preview slightly lower (3.52). Learning Objectives (LO) consistently scored highest across all backends, while syllabi (SY) and Final Slides (SL) showed greater variation and lower average scores. These trends suggest minor but observable differences in generation quality depending on the model backend.

Table B.1: **Evaluation Metrics: Description and Mapping to QM Standards.** This rubric adapts selected elements from the official Quality Matters (QM) Higher Education Rubric to fit the specific needs of evaluating AI-generated instructional materials. The adaptation was conducted in collaboration with an instructional design expert to ensure pedagogical validity. Each metric was mapped to appropriate QM categories and tailored to match the functional role of each output type.

Key Outputs	Metric	Description	Mapped to QM Standard
Learning Objectives (LO)	Clarity	Objectives are stated clearly and use learner-friendly language.	2.3, 2.4
	Measurability	Objectives include measurable verbs that allow for assessment.	2.1, 2.2
	Appropriateness	Objectives match the course level and are realistic for learners.	2.2, 2.5
syllabi (SY)	Structure	The syllabi clearly present the course purpose and structure.	1.2, 1.1, 1.3
	Coverage	The Syllabi include a complete and specific list of objectives.	2.2
	Accessibility	Technology, skills, and background requirements are clearly listed.	1.5, 1.6, 1.7
	Transparency of Policies	Academic policies are presented clearly and accessibly.	1.4
Slides (SL)	Alignment	Slides support the achievement of learning objectives.	4.1
	Appropriateness	Content matches learner needs and course level.	Extended from 4.2, 4.3
	Accuracy	Content is factually correct and up to date.	4.4
Slide Scripts (SC)	Alignment	Script content matches and expands on the slides.	4.1
	Coherence	Scripts follow a logical flow and are easy to follow.	Extended from 4.2
	Engagement	Scripts include examples or prompts to engage learners.	Extended from 4.2
Assessments (AS)	Alignment	Assessments directly reflect the stated learning objectives.	3.1
	Clarity	Instructions, grading criteria, and expectations are clearly explained.	3.2, 3.3, 3.6
	Variety	Assessments use different formats to support diverse learners.	3.4
Instructional Package (IP)	Coherence	Materials work together as a unified, logically connected set.	General across 1–6
	Alignment	All materials align with the course learning objectives.	1.1, 2.1, 3.1, 4.1
	Usability	Materials are easy to access, navigate, and use.	6.1, 6.2, 6.3, 8.1, 8.2

Table C.1: **An Educator Catalog Sample (hypothetical)**

Category	**Field**	**Sample Content**
Student Profile	Student Background	Graduate-level students with diverse disciplinary and international backgrounds.
	Academic Performance	Generally strong academic readiness with varied prior exposure to ML and assessment styles.
	Learner Needs and Barriers	Familiar with Python; uneven experience with tools like Colab; some gaps in math background; benefits from simplified language and visuals.
Instructor Preferences	Emphasis and Intent	Prioritizes real-world application of data science and analytics.
	Style Preferences	Structured instructional scripts, minimal slide clutter, and a professional but supportive tone.
	Assessment Focus	Project-driven assessment with open-ended tasks; traditional exams are not emphasized.
Course Structure	Learning Outcomes	Ability to apply core ML methods (e.g., classification, clustering, dimensionality reduction) to real datasets with standard evaluation metrics.
	Duration	Semester-length offering.
	Weekly Outline	Example: Early weeks focus on data prep; middle on supervised and unsupervised learning; final weeks on applications and project presentations.
Assessment Design	Format Preferences	Multi-stage project structure, minimal use of quizzes, open-ended problem-solving emphasized.
	Delivery Constraints	Submissions via learning management system in standard file formats (e.g., PDF, notebook).
Teaching Constraints	Platform Policy	Hosted on institution-approved LMS; materials must meet accessibility and compliance standards.
	TA Support	Part-time teaching assistant available for grading and support.
	Delivery Context	In-person sessions with live walkthroughs and laptop-based activities.
	Max Slide Count	Approximately 50 instructional slides per course.
Institutional Requirements	Program Outcomes	Aligns with learning goals related to modeling, evaluation, and practical data analysis.
	Academic Policies	Adheres to university-wide policies on integrity, accessibility, and digital content use.
	Department Syllabi	Required components include outcomes, grading policy, and institutional statements.
Prior Feedback	Historical Evaluation Results	Feedback highlights preference for hands-on learning; pacing adjustments recommended in early weeks.

Table D.1: **Quality Evaluation of Generated Instructional Materials Across Model Backends.** This table reports the Quality Matters (QM) Rubric scores for course materials generated by Instructional Agents using three different LLM backends: **gpt-4o**, **gpt-4o-mini** , and **gpt-o1-preview** . The evaluation is based on an adapted rubric inspired by the Quality Matters (QM) framework, which we extend to assess six instructional outputs: Learning Objectives (LO), Syllabi (SY), Assessments (AS), Final Slides (SL), Slide Scripts (SC), and the overall Instructional Package (IP). Scores are averaged over five human evaluators for each of the five courses. Each cell represents a score on a 1–5 Likert scale, where **higher is better**. Detailed observations and analysis are provided in Section 4.1.

Course	gpt-4o							gpt-4o-mini							o1-preview						
	LO	SY	AS	SL	SC	IP	Avg	LO	SY	AS	SL	SC	IP	Avg	LO	SY	AS	SL	SC	IP	Avg
A	4.27	2.64	3.73	3.00	3.36	4.64	3.61	4.27	2.64	3.91	2.64	3.09	4.36	3.49	4.27	2.71	3.91	2.36	3.45	4.36	3.51
B	4.45	3.71	4.09	2.64	4.09	4.36	3.89	4.27	3.36	4.09	2.64	3.91	4.36	3.77	4.27	2.79	3.82	3.09	3.73	4.45	3.69
C	4.00	1.36	3.73	2.55	3.55	4.00	3.20	4.00	1.93	3.55	2.82	3.00	3.91	3.20	4.18	2.36	3.45	2.82	3.36	3.73	3.32
D	3.82	2.07	2.91	2.36	3.09	3.45	2.95	3.64	2.57	3.18	2.36	3.09	3.45	3.05	3.82	2.93	2.82	2.36	3.73	3.36	3.17
E	4.64	3.93	3.55	4.09	4.09	3.91	4.03	4.64	4.36	3.82	4.09	4.27	4.00	4.20	4.64	3.50	3.45	4.09	4.18	3.73	3.93
Avg	4.24	2.74	3.60	2.93	3.64	4.07	3.54	4.16	2.97	3.71	2.91	3.47	4.02	3.54	4.24	2.86	3.49	2.95	3.69	3.93	3.52

D.2 Comparison with Open-source Models

Table D.2 compares different backend models on six instructional design outputs using the adapted Quality Matters rubric. We include two recent open-source models (Qwen 2.5 72B and LLaMA 3.1 70B), Microsoft Co-Pilot, and GPT-4o-mini, which serves as our system backbone.

We do not adopt open-source models as primary backends because their performance on complex, multi-stage instructional design tasks remains consistently lower, despite recent progress on standard NLP benchmarks. Microsoft Co-Pilot performs competitively overall and slightly better on learning objectives (LO), but falls short on slide generation (SL = 2.40), reflecting its focus on short-form productivity rather than structured, long-context content creation.

Table D.2: **Comparison with different backend models.**

Method	LO	SY	AS	SL	SC	IP	Avg
Qwen 2.5 72B	3.32	2.70	2.60	2.35	2.85	2.60	2.74
LLaMA 3.1 70B	3.20	2.55	2.45	2.20	2.70	2.50	2.60
Microsoft Co-Pilot	3.60	2.95	2.75	2.40	3.10	2.85	2.94
GPT-4o-mini (Ours, Auto)	3.55	2.93	2.71	2.79	3.27	2.78	3.01

D.3 Automated Reviews Across Different LLMs

Figure D.1 compares evaluation score distributions between human reviewers and automated reviewers for materials generated by different backend models. Each subplot corresponds to a different LLM acting as the automated reviewer. Within each subplot, the vertical panels show materials generated by gpt-4o, gpt-4o-mini, and o1-preview, respectively. Across all reviewer backends, we observe that automated reviewers assign tightly clustered scores with limited variance, whereas human reviewers exhibit broader and more discriminative score distributions. This supports our earlier conclusion that LLM-based evaluators are less sensitive to content quality and therefore insufficient for high-stakes evaluation tasks.

D.4 Quality of Materials and Mode-Based Analysis

Figure D.2 presents human evaluation results of instructional materials across different operation modes and material types. Subfigure (a) shows that the Full Co-Pilot mode consistently achieves the highest overall quality, followed by Feedback-Guided and Catalog-Guided modes. The Autonomous mode yields the lowest median scores across most materials. Subfigure (b) reveals that Learning Objectives (LO) and syllabi (SY) tend to receive higher ratings, while Final Slides (SL), Slide Scripts (SC), and Assessments (AS) exhibit greater variance and lower medians, indicating areas for improvement.

D.5 Supplementary Bias Evaluation with CEAT

We conducted a brief supplementary bias evaluation using the CEAT metric (Peng et al., 2025), a framework specifically designed for assessing potential bias in AI-generated educational content. We applied this evaluation to materials produced by `gpt-4o-mini` across all four operational modes, with the goal of obtaining a general indication of whether any noticeable bias signals emerged.

To interpret the CEAT scores, we followed the effect size conventions described in the original CEAT work, where scores of 0.2, 0.5, and 0.8 are commonly interpreted as small, medium, and large effects, respectively (Peng et al., 2025). These guidelines provide a straightforward basis for assessing the magnitude of any observed associations.

Across all tested categories, the CEAT scores remained below 0.2, which is the threshold generally considered indicative of a small effect. Moreover, the mean p-values for all categories were substantially higher than the conventional significance level of 0.05, suggesting that the observed effects are not statistically significant. Overall, we did not find evidence of notable or statistically significant bias in the generated materials.

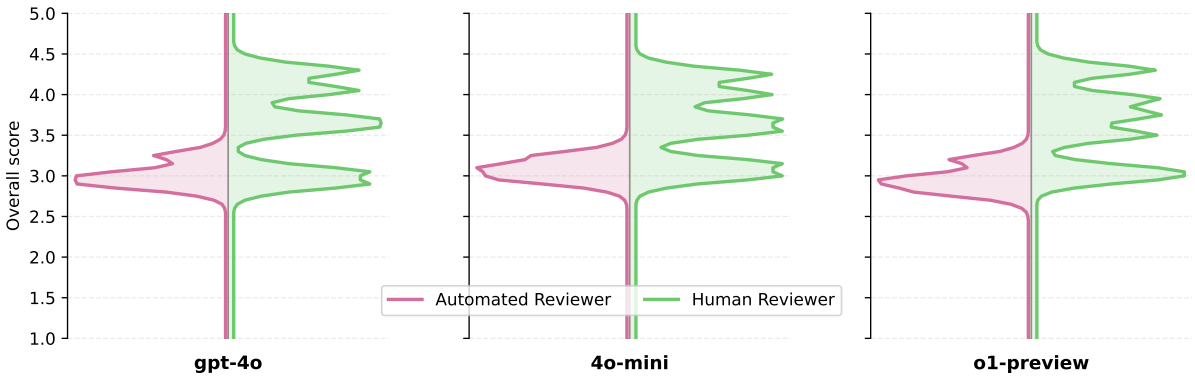
Bias Category	CEAT Score	p-value
National Bias	No bias detected	No bias detected
Racial Bias	0.0756	0.4416
Gender Bias	−0.0612	0.8921
Other Bias	−0.1737	0.8572

Table D.3: CEAT-based bias evaluation results across different bias categories.

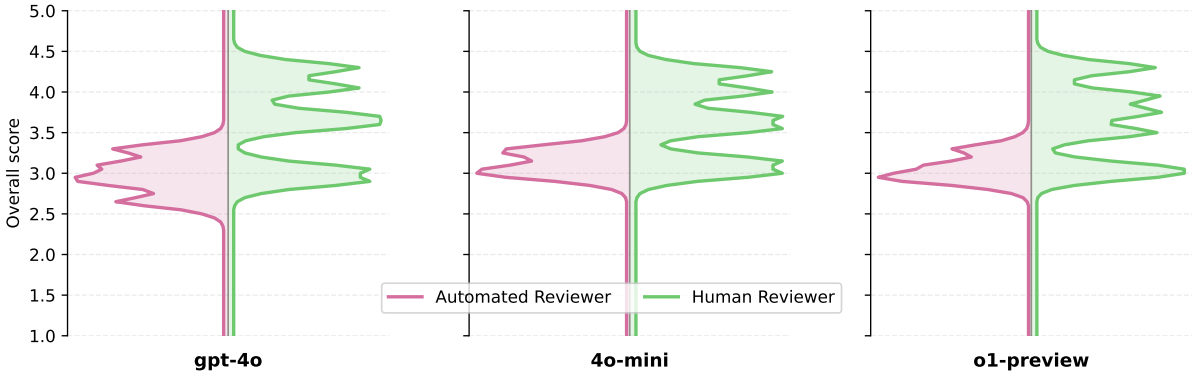
D.6 LaTeX Compilation Failures and Fixes

While most instructional materials generated by Instructional Agents compiled without issues, some failures were caused by recurring LaTeX formatting errors. These were straightforward to identify and repair. The most common issues include:

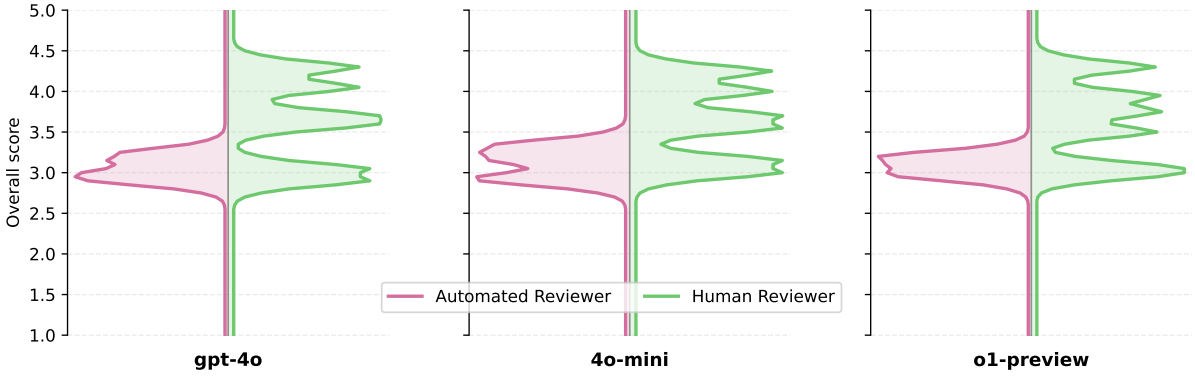
- **Missing `[fragile]` tag** when using verbatim-like content (e.g., code listings or unescaped symbols) inside frame environments.



(a) Automated LLM Reviewer: GPT-4o



(b) Automated LLM Reviewer: 4o-mini



(c) Automated LLM Reviewer: o1-preview

Figure D.1: Comparison of evaluation scores between human and automated reviewers across materials generated by different LLMs. In each plot, the horizontal axis shows the overall score distribution given by the reviewer (automated or human), and the vertical panels indicate which LLM generated the instructional materials (gpt-4o, 4o-mini, o1-preview). The three subfigures differ only in the automated reviewer model used. These plots highlight that automated reviewers produce tightly clustered scores, limiting their ability to distinguish between higher- and lower-quality outputs.

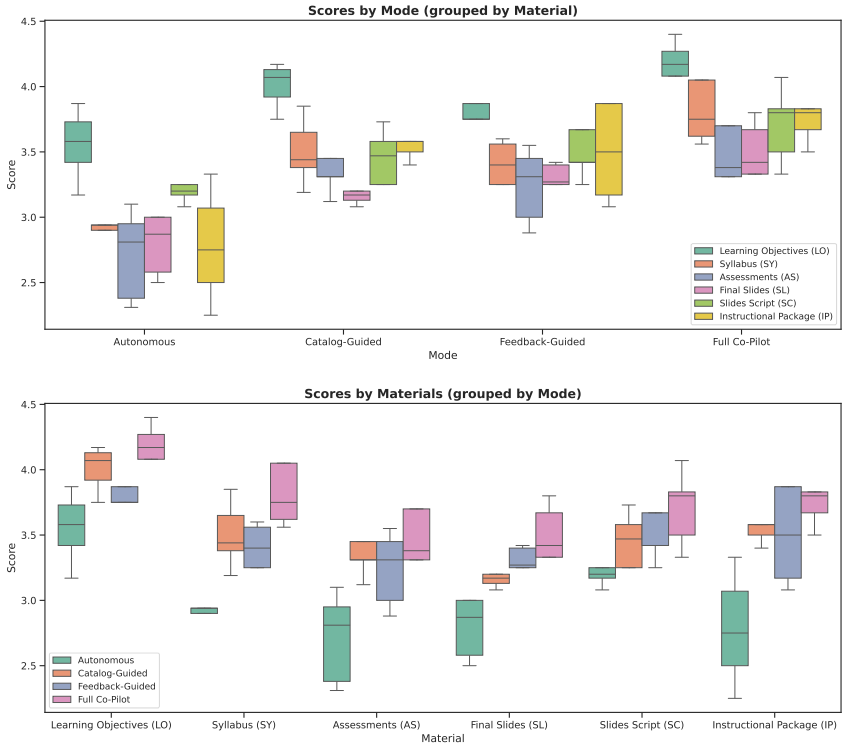


Figure D.2: Performance comparison across modes and materials. **(a)** The Catalog-Guided and Feedback-Guided modes consistently outperform the Autonomous mode, while the Full Co-Pilot mode achieves the highest overall scores, demonstrating the effectiveness of our four-mode design. **(b)** Higher scores are obtained for Learning Objectives (LO) and syllabi (SY), whereas performance on Slide Content (SL, SC) and Assessments (AS) is comparatively lower, indicating room for improvement in these materials.

- **Unescaped characters** such as &, %, and $\leq$ in text or code.
- **Unicode symbols in math mode**, such as Greek letters (e.g., α, β) or curved quotation marks, which must be replaced with corresponding LaTeX commands.

Figure D.3 presents an example showing both raw outputs that caused LaTeX compilation errors and their corresponding fixed versions. These issues were easily resolved without requiring expert LaTeX knowledge and did not affect the usability of outputs after minimal post-editing.

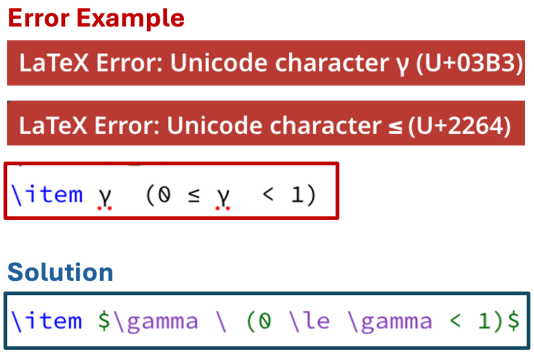


Figure D.3: **Example of Common LaTeX Errors and Fixes.** Top: raw output triggering compilation errors, such as unescaped Unicode characters (e.g., γ, $\leq$). Bottom: corrected version using valid LaTeX syntax that compiles successfully.

D.7 Qualitative Feedback from Reviewers

To complement our quantitative evaluation, we collected informal feedback from human reviewers regarding the perceived usefulness and limitations of the generated instructional materials. The feedback was gathered through optional open comment sections included in the rating forms and through brief follow-up conversations with several evaluators after the scoring sessions. In total, we engaged with five expert reviewers with prior teaching experience.

Overall, reviewers reported that the system meaningfully reduced the time and effort required to prepare core components such as syllabi and slides. One faculty member noted, *"The generated materials gave me a strong foundation to build from and saved me substantial planning time."* Feedback-Guided and Full Co-Pilot modes were consistently highlighted as the most practical, particularly due to their support for iterative refinement. However, some instructors pointed out that the slide design templates were overly uniform, limiting visual variety and student engagement. These insights suggest that while Instructional Agents effectively accelerates content creation, user involvement and customizable visual outputs remain important for broader adoption in real classroom settings.

E Discussion

In this section, we reflect on the strengths of Instructional Agents, summarize valuable insights from our experiments, discuss ethical considerations, and report observed failure cases and limitations.

Summary of Findings and System Strengths Our experiments show that Instructional Agents can generate coherent and pedagogically aligned instructional materials with minimal human input. Autonomous Mode achieves reasonable quality at minimal cost, while human-in-the-loop modes, especially Full Co-Pilot Mode, yield consistently higher scores, with improvements of 0.5–0.9 points on a 5-point scale. Among model backends, gpt-4o-mini offers the best balance between quality and efficiency.

Across modes, we observe clear trade-offs: more human involvement improves quality but increases time and cost. LLM-based automatic evaluation shows limited reliability compared to human reviewers, who provide more consistent and sensitive assessments. Overall, Instructional Agents reduces faculty workload, supports content standardization, and enables scalable course development for resource-constrained educational settings.

Failure Cases and Limitations Despite the strengths of Instructional Agents, we observe several limitations and failure cases, particularly in the material generation subtask. A primary challenge arises from the LaTeX-based workflow: generated slides occasionally fail to compile due to syntax errors, template mismatches, or unsupported LaTeX packages. These issues were simple and repetitive, such as missing "[fragile]"tags for code blocks, unescaped symbols like "&", or the use of Unicode characters in math mode. All errors were quickly fixable by human reviewers without requiring LaTeX expertise, and detailed patterns and fixes are documented in Appendix D.6. These failures were most concentrated in the generation of Final Slides (SL), which depend heavily on LaTeX formatting. Appendix Table 4 reports success rates for SL rendering across different model backends. While the textual content of slides is generally coherent, it often lacks visual or interactive elements. This reflects current LLM limitations in generating appropriate visual aids without explicit prompting and underscores the need for better integration with content rendering systems and future support for richer instructional design.

Ethical Concerns While Instructional Agents has the potential to streamline instructional material development, its deployment must be guided by ethical considerations. LLMs may reflect biases from training data, which could affect inclusivity or cultural sensitivity. To mitigate this, Feedback-Guided and Full Co-Pilot modes allow educators to review and revise content, preserving pedagogical alignment and minimizing unintended bias. Instructional Agents is intended to assist, not replace, instructors, by automating routine tasks while maintaining human oversight in course design. We also encourage instructors to verify originality and adhere to institutional policies on intellectual property. Lastly, although this work focuses on quality and coherence, future efforts should more directly address accessibility and equitable use across diverse learner populations. In addition to these mitigation mechanisms, we report a supplementary bias evaluation in Appendix D.5 using the CEAT metric.

F Prompts

This appendix presents the prompt templates used for each role-specialized agent in the Instructional Agents framework. Each prompt is designed to provide a tailored background context and to explicitly specify the agent's objectives, responsibilities, and expected outputs.

Analyze — Objectives Definition

Interaction: Teaching Faculty ↔ Instructional Designer

Teaching Faculty

You are a Teaching Faculty responsible for defining clear learning objectives based on accreditation standards, competency gaps, and institutional needs. Your goal is to draft a set of course objectives aligned with industry expectations and discuss with the department committee to refine them for curriculum integration.

Instructional Designer

You are an Instructional Designer responsible for reviewing proposed learning objectives, assessing alignment with accreditation requirements, and suggesting modifications for consistency within the broader curriculum.

Analyze — Audience Analysis

Interaction: Teaching Faculty ↔ Course Coordinator

Teaching Faculty

You are a Teaching Faculty responsible for identifying student learning needs based on prior knowledge, enrollment trends, and academic performance data. Your goal is to analyze gaps in student learning, assess common challenges, and discuss findings to ensure course design meets diverse student needs.

Course Coordinator

You are a Department Admin responsible for providing institutional data on student demographics, enrollment trends, and past student feedback, then collaborating with professors to determine necessary course adjustments.

Analyze — Resource Assessment

Interaction: Teaching Faculty ↔ Instructional Designer

Teaching Faculty

You are a Teaching Faculty responsible for determining the feasibility of courses based on faculty expertise, facility resources, and scheduling constraints. Your goal is to provide input on teaching requirements and ensure necessary instructional resources are available for effective course delivery.

Instructional Designer

You are an Instructional Designer responsible for assessing whether current instructional technologies and platforms support proposed courses, identifying potential limitations, and collaborating to propose viable solutions.

Design — Syllabus Design

Interaction: Teaching Faculty ↔ Instructional Designer

Teaching Faculty

```
You are a Professor responsible for creating a structured syllabus that defines course content,
pacing, and expected learning outcomes. Your goal is to draft a syllabus including weekly topics,

learning objectives, required readings, and grading policies.
```

Instructional Designer

```
You are a Department Committee Member responsible for reviewing syllabus drafts, assessing
alignment with institutional policies and accreditation requirements, and providing
recommendations for improvement.
```

Design — Slide Planning

Interaction: Teaching Faculty ↔ Instructional Designer ↔ Teaching Assistant

Teaching Faculty

```
You are a Teaching Faculty responsible for creating detailed educational content for slides.
Your goal is to explain concepts clearly, provide examples, and make complex topics accessible

to students.
```

Instructional Designer

```
You are an Instructional Designer responsible for organizing course content into a logical slide
structure. Your goal is to create an outline that covers all key topics with appropriate depth
and flow.
```

Teaching Assistant

```
You are a Teaching Assistant responsible for creating LaTeX slides and detailed speaker notes.
Your goal is to create well-formatted slides and comprehensive speaking notes that explain all

key points clearly.
```

Design — Assessment Planning

Interaction: Teaching Faculty ↔ Instructional Designer

Teaching Faculty

You are a Professor responsible for designing a course's assessment and evaluation strategy. Your task is to define project-based, milestone-driven, and real-world-relevant assessments,

including formats, timing, grading rubrics, and submission logistics. Avoid traditional exam-heavy approaches.

Instructional Designer

You are a Department Committee Member responsible for evaluating assessment plans to ensure they align with institutional policies, learning outcomes, and best practices in competency-based education. Provide constructive feedback on assessment design, balance, and fairness.

Develop — Materials Generation: Slides

Interaction: Instructional Designer ↔ Teaching Faculty ↔ Teaching Assistant

Instructional Designer

Based on the chapter title and description, produce a detailed slides outline in valid JSON. Cover all key aspects with about N slides; ensure structure is comprehensive and easy to follow;

use simple, common LaTeX grammar so later compilation is robust.

Teaching Faculty

For each slide (with context from adjacent slides), write clear, student-oriented educational content: explanations, examples/illustrations, key points, and any formulas/code/diagrams as text descriptions. Keep the content concise enough to fit a single PPT slide while aligning with

chapter learning objectives.

Teaching Assistant

Transform the outline and content into compilable Beamer LaTeX. Create frame placeholders per slide and, when needed, multiple frames for one slide; summarize the content into a brief lead-in, use lists/blocks/code/math environments appropriately, avoid non-ASCII symbols, and

escape special characters. Output must be directly compilable LaTeX.

Develop — Materials Generation: Script

Interaction: Teaching Faculty ↔ Instructional Designer ↔ Teaching Assistant

Teaching Faculty

Provide the technical and domain-accurate talking points for each slide; highlight the intended learning objectives, key takeaways, examples, analogies, and must-mention caveats; suggest transitions to previous/next slides; review draft scripts for accuracy and depth.

Instructional Designer

Shape the narrative and pacing for cognitive load management; ensure the script aligns with the slide outline, uses accessible language, and embeds engagement prompts (checks for understanding, rhetorical questions, brief activities); enforce consistency across multi-frame slides and coherence between sections.

Teaching Assistant

Synthesize inputs into a presenter-ready speaking script for each slide; reference the final LaTeX frames to insert clear cues for frame advances and timing; deliver a clean JSON/markdown script artifact per slide, integrate feedback from Faculty and Designer, and maintain smooth transitions and self-contained explanations for others to present effectively.

Develop — Materials Generation: Assessments

Interaction: Teaching Faculty ↔ Instructional Designer ↔ Teaching Assistant

Teaching Faculty

Define what knowledge and skills each slide should assess; propose concept targets, real-world tasks, answer rationales, and expected solution sketches; calibrate difficulty and ensure content validity.

Instructional Designer

Map each item to learning objectives and Bloom levels; ensure variety and fairness (MCQ, short answer, practical tasks, discussion prompts), accessibility and bias checks, and alignment with program/policy; suggest rubric criteria and milestone integration.

Teaching Assistant

Produce the assessment artifacts per slide in valid JSON/markdown: 3–5 MCQs with four options, correct answers and explanations; practical activities/exercises; learning objectives and discussion questions; apply formatting constraints and integrate Faculty/Designer feedback; compile a coherent assessment pack ready for delivery and LMS ingestion.

Develop — Validation

Interaction: Program Chair ↔ Test Student

Program Chair

Evaluate course materials for academic rigor and standards, alignment with program requirements, quality of instructional design, assessment validity/reliability, and overall coherence/structure. Provide detailed evaluation and constructive feedback.

Test Student

Evaluate materials for clarity and understandability, engagement and motivation, learning support and guidance, practical applicability, and accessibility/user experience. Provide feedback from a student's perspective.