

Article

A Systemic Intervention for Human–Artificial Intelligence Co-Design of Lesson Plans: Integrating Pedagogical Theories into Prompt Engineering

Yinan Lu ¹ , Weinuo Li ² and Yue Cai ^{1,*} 

¹ Institute of Vocational and Technical Education, Tongji University, Shanghai 201804, China; luyinan915@tongji.edu.cn

² College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China; liweinuo@tongji.edu.cn

* Correspondence: caiyue@tongji.edu.cn

Abstract

Large language models (LLMs) can assist lesson planning, but simple prompts often yield incomplete and misaligned outputs. This study proposes a three-step prompt framework grounded in Bloom’s taxonomy, Adaptive Control of Thought-Rational (ACT-R) theory, Gagné’s nine events, and problem-chain theory, decomposing planning into objective, unit, and activity design. Using three DeepSeek models (R1, V3, 32B) and five prompting strategies, 150 lesson plans were generated on ten computer networking topics. Coverage of Gagné’s nine events and functional quality were evaluated via an LLM judge and human validation. All theory-based strategies significantly outperformed naive prompting, raising Gagné’s event coverage above 90% in the full corpus and from 74.1% to 89.8–93.5% in human ratings. Functional quality scores improved by up to 17.3% (LLM judge) and 53.6% (human raters). Gagné’s five-stage design outperformed ACT-R’s three-stage design under base conditions, while problem-chain guidance benefited ACT-R substantially. Model capability moderated gains: smaller models benefited most in structural completeness, stronger reasoners achieved higher absolute quality. These findings demonstrate that pedagogically grounded, multi-stage prompts are designed to reconfigure teacher-artificial intelligence (AI) interaction from passive output consumption toward structured collaborative design, offering a scalable intervention for integrating LLMs into instructional workflows.

Keywords: lesson planning; pedagogical frameworks; prompt engineering; LLMs; human–AI collaboration



Academic Editors: Xuesong Zhai, Andreja Istenic and Jiong Guo

Received: 27 May 2026

Revised: 12 July 2026

Accepted: 15 July 2026

Published: 16 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

AI is transforming education, but most AI applications target online learning platforms, intelligent tutoring systems, and automated assessment tools. Traditional face-to-face classroom teaching, which remains the dominant mode of instruction from primary to higher education, has received less attention from the AI community [1]. Within traditional classrooms, lesson planning is a critical task that directly determines teaching quality and student learning outcomes. However, designing high-quality lesson plans requires deep integration of pedagogical knowledge, subject matter expertise, and an understanding of student cognition. This complexity makes lesson planning challenging, particularly for novice teachers [2].

LLMs have recently demonstrated remarkable capabilities in text generation, instruction following, and complex reasoning [3]. These abilities open new possibilities for assisting teachers in lesson planning. Early studies show that LLMs can generate lesson plans when given simple prompts such as “design a lesson plan about photosynthesis” [4]. Nevertheless, mounting evidence indicates that naive prompting produces outputs that are functionally incomplete, structurally unstable, and often misaligned with actual teaching needs [5]. Research has documented various deficiencies, including missing components such as learner analysis, unstable structural consistency across generations, and difficulties in integrating disciplinary culture and cross-disciplinary assessment [4–8]. These limitations point to a central problem: LLMs do not intrinsically understand what constitutes a pedagogically sound lesson plan. Their outputs are probabilistic reconstructions of training data patterns, not principled applications of educational theory.

To address these issues, this study proposes a three-step prompt engineering framework for human–AI collaborative lesson planning. Unlike prior structured-prompt approaches that generate lesson plans in a single pass [2,6,8], this framework decomposes planning into three sequentially dependent subtasks—teaching objective design, instructional unit design, and activity design. Each subtask is guided by specific pedagogical theories, including Bloom’s taxonomy, ACT-R cognitive theory, Gagné’s nine events, and problem-chain teaching theory. The framework allows teachers to intervene at any intermediate step, modifying outputs and thus steering subsequent generation. It also enables a systematic comparison of how different pedagogical choices contribute to plan quality. This design is intended to transform teachers from passive recipients of model outputs into more active participants in the design process, as they can review and modify intermediate outputs before they feed into subsequent steps. An evaluation using three LLMs demonstrates that the framework significantly improves instructional event coverage and functional quality.

2. Literature Review

2.1. AIED Support for Teachers in Traditional Classrooms

The broader field of artificial intelligence in education (AIED) has explored ways to support teachers in traditional classroom settings. Standen et al. [9] utilized a multi-modal sensor-based AI system to recognize students’ emotional states, assisting teachers in optimizing content presentation and communication strategies. Martinez-Maldonado et al. [10] developed a multi-touch tabletop system that provides teachers with real-time analytics of small group work. Fu et al. [11] designed an automatic scoring tool using speech recognition, enabling immediate pronunciation feedback. These studies demonstrate AI’s potential to support teachers, but they focus on behavior detection, assessment, or classroom management rather than the core task of lesson planning.

Zawacki-Richter et al. [12] conducted a systematic review of AI applications in higher education and found that teacher-facing tools remain rare, with most research concentrating on student-facing applications such as intelligent tutoring systems and personalized learning. Forero-Corba and Bennisar [13] similarly noted that only a small fraction of AIED studies address teacher professional development or classroom orchestration. Brusilovsky [14] further observed that human–AI collaboration research in education has lagged behind other AI application domains, with few frameworks supporting teachers as active partners rather than passive recipients of AI outputs. This gap indicates that while AI has been deployed to assist teachers in peripheral tasks, the central instructional design process, especially lesson planning, has received limited support from the AIED community.

2.2. Large Language Models in Lesson Planning

Several recent studies have examined the use of LLMs for automated lesson planning. Davis and Lee [4] found that ChatGPT-generated lesson plans often lack logical progression between modules and fail to integrate mathematical culture. Hu et al. [6] reported that while GPT-4 performs well in setting instructional objectives, it struggles with cross-disciplinary assessment and the integration of domain-specific cultural contexts. Karaman and Goksu [7] conducted a comparative experiment and demonstrated that lesson plans produced by ChatGPT-4.0 do not surpass those designed by experienced teachers, with concerns remaining about information accuracy and ethical issues. Zheng et al. [8] further identified that naive prompting leads to missing components such as learner analysis and unstable structural consistency across repeated generations. They proposed a three-stage generation framework incorporating self-critique and retrieval mechanisms, which improved output quality but introduced additional complexity. Naveed et al. [15] provided a comprehensive overview of LLMs, noting that zero-shot prompting achieves only limited accuracy in complex tasks, while few-shot prompting improves performance at the cost of increased prompt design effort.

These limitations point to a common problem: LLMs do not intrinsically understand the pedagogical logic underlying a sound lesson plan. Their outputs are probabilistic reconstructions of training data, not principled applications of educational theory.

2.3. Prompt Engineering for Pedagogical Enhancement

Prompt engineering refers to the design of task-specific instructions that activate a LLM's internal knowledge and shape its behavior without modifying model parameters [16]. Unlike traditional fine-tuning, which requires retraining or adjusting model weights, prompt engineering works by embedding natural language directives, examples, or reasoning chains into the input. This approach is particularly valuable in domains where labeled data are scarce and model interpretability is important [17].

A landmark advance in prompt engineering is chain-of-thought (CoT) prompting, proposed by Wei et al. [18] CoT prompting decomposes complex reasoning problems into a series of intermediate steps, explicitly showing the model how to arrive at an answer. This technique has been shown to significantly improve performance on arithmetic, common-sense, and symbolic reasoning tasks. Other prompt engineering strategies include zero-shot prompting, where the model receives only a task description without examples, and few-shot prompting, where a few input-output pairs are provided as demonstrations [19].

In the domain of lesson planning, researchers have applied structured prompts to compensate for LLMs' lack of intrinsic pedagogical knowledge. Fan et al. [2] developed an interactive system called LessonPlanner that guides novice teachers through generating lesson plans based on Gagné's nine events. Their user study showed that structured prompts grounded in pedagogical theory improve the completeness of lesson plans, but teachers remain largely passive recipients of model outputs. van den Berg and his colleague [5] explored the possibilities of ChatGPT for lesson planning and emphasized the need for critical thinking and openness in teacher education, noting that prompt design requires substantial expertise. Elsayary et al. [20] examined teachers' use of prompt-engineered GenAI tools for lesson planning and found that well-designed prompts improve both efficiency and content relevance, while highlighting the need for ongoing teacher AI literacy support.

In educational applications, structured prompts that incorporate pedagogical theories have consistently been shown to produce more complete and reliable outputs compared to naive prompts [2,6,8]. Peikos and Stavrou [21] showed that embedding pedagogical content knowledge (PCK) elements into prompts improved the quality of ChatGPT-generated science lesson plans, providing early evidence that pedagogical theory can structure LLM out-

puts toward more coherent instructional designs. Bai et al. [22] included the 5E instructional model into an AI chatbot for scaffolding instructional design and found that combining theory-informed AI support with human teacher feedback outperformed either approach alone, reinforcing the value of theory-driven human–AI collaboration. Zheng et al. [8] extended prompt engineering by incorporating self-critique and retrieval-augmented generation into a three-stage framework. Their approach improved the logical coherence of generated lesson plans but increased the technical burden on teachers. Hu and colleagues [23] introduced a teaching simulation approach in which an LLM-generated lesson plan is evaluated by a simulated classroom and iteratively refined based on the feedback. The approach still treats lesson planning as an end-to-end generation task, as teachers can only judge the final output without the ability to intervene at intermediate design stages. Cai et al. [24] found that ChatGPT-supported lesson plan critiques enhanced pre-service teachers’ planning skills, though the interaction remained feedback-oriented rather than enabling teachers to steer the generation process itself.

2.4. Research Gap and Question

In summary, the literature reveals that most existing AIED systems for traditional classrooms focus on behavior detection or assessment, not on supporting the core task of lesson planning. While LLMs have shown promise, existing prompt engineering approaches do not allow fine-grained teacher intervention. Teachers can only adjust final outputs rather than guide the generation process at multiple levels. In addition, although individual pedagogical theories such as ACT-R and problem-chain theory have been proven effective in classroom practice, they have not been systematically integrated into a single prompt engineering framework. Tian [25] proposed a conceptual human–AI collaborative instructional design model grounded in activity theory, but the model has not been empirically tested with concrete prompt engineering strategies.

To address these gaps, this study proposes a three-step human–AI collaborative lesson planning framework that decomposes the design process into structured sub-tasks and grounds each sub-task in established pedagogical theories. The framework is designed to allow teachers to intervene at any intermediate step, transforming them from passive recipients of model outputs into genuine co-designers. The effectiveness of this framework is then investigated through the following research questions:

RQ1. *Does the proposed framework systematically enhance the completeness and quality of LLM-generated lesson plans compared to naïve prompting?*

RQ2. *How do different pedagogical design choices within the framework contribute to the improvements?*

RQ3. *How does LLM capability, in terms of model scale and reasoning strength, moderate the effectiveness of the framework?*

3. Pedagogically Grounded Human–AI Collaborative Lesson Planning Framework

To answer the research questions outlined in Section 2.4, this section presents a pedagogically grounded framework for human–AI collaborative lesson planning. The framework addresses the core limitation identified in the literature by decomposing lesson planning into three sequentially dependent sub-tasks, each guided by a specific educational theory. This decomposition transforms the teacher’s role from passive recipient of LLM outputs to active co-designer who can review and modify intermediate results at each step before they feed into the next.

3.1. Definition of Lesson Planning

In this study, lesson planning refers to the design of a single-session instructional plan that specifies the teaching flow and activity organization and serves as the direct basis for classroom instruction. The quality of classroom teaching depends largely on the quality of this plan.

Designing a high-quality classroom plan is a complex, multi-dimensional task. The designer must integrate knowledge of students' current cognitive development levels, embed the core competencies of the subject, and accurately target the knowledge system of the lesson. Furthermore, the design process needs to incorporate insights from cognitive science and educational theory, such as Bloom's taxonomy and constructivist principles. This complexity makes lesson planning a challenging creative endeavor.

For LLMs, the instruction "design a lesson plan" is an abstract concept that lacks a clear operational path. When given a single simple prompt, a LLM does not generate a plan based on systematic pedagogical reasoning. Instead, it retrieves and recombines text fragments associated with keywords such as "lesson design" from its training corpus, producing an output that is essentially a conditional sample of the training data distribution [8]. Therefore, even models with strong text understanding and instruction-following capabilities cannot easily produce teacher-satisfactory lesson plans through naive prompting.

To improve the quality and usability of LLM generated lesson plans, prompt design and optimisation are of central importance. A systematic review of LLM applications in instructional planning found that effectiveness is heavily conditioned by prompt quality, with poorly designed prompts identified as a primary barrier to usable outputs [26]. More broadly, Qian's review of prompt engineering across educational contexts concluded that technique-based and process-based prompt designs consistently outperform generic prompting, confirming that prompt engineering is not merely a technical convenience but a determinant of pedagogical quality [27]. Within lesson planning, Zheng et al. [8] demonstrated that a structured three-stage framework incorporating retrieval-augmented generation and self-critique significantly improves logical coherence and applicability, while Hu et al. [6] showed that embedding pedagogical content knowledge and Bloom's taxonomy into prompts yields more cognitively coherent objectives and activity designs. These findings indicate that the shift from naive to theory-grounded prompting is not an incremental refinement but a necessary condition for producing usable lesson plans.

The design of teaching plans is a complex task involving multiple elements, which requires an in-depth exploration of its internal logic and systematic optimization [28]. Based on the basic principle of task decomposition [29], this article breaks down this complex task into three interrelated and progressively nested sub-tasks:

- (1) Teaching objective design. Teaching objectives form the foundational step of the entire planning process [30]. As the high-level instructional guideline, they specify the cognitive outcomes students are expected to achieve by the end of the lesson. Each objective combines a specific cognitive level (e.g., understanding, applying, analyzing from Bloom's taxonomy) with a concrete knowledge point, resulting in an observable and assessable target. Designing these objectives requires aligning with the core purpose of the course, matching students' current cognitive development, embedding the subject's essential competencies, and anchoring the logical nodes of the domain's knowledge structure. The quality of this step directly determines the direction and effectiveness of all subsequent design work.
- (2) Instructional unit design. This step divides the whole lesson into several relatively independent yet sequentially connected units, each organized around a distinct thematic focus. Some existing approaches [2,8] only include objective design and final activity generation, omitting this intermediate phase. Such oversimplification ignores

the intrinsic logic of instruction: moving from objectives to specific activities requires a structured, transitional layer. In authentic teaching practice, teachers first extract core themes from the objectives, then partition the lesson into progressive units, and only afterwards design detailed activities for each unit. Decomposing lesson planning to include this separate unit-design step respects the natural human cognitive progression and the hierarchical nature of knowledge transmission [31]. It prevents the content fragmentation or logical gaps that often occur when jumping directly from objectives to activities, thereby ensuring systematic coherence and continuity throughout the lesson.

- (3) Teaching activity design. It assigns a concrete implementation path to each previously defined instructional unit. Because the overall lesson logic and the focus of each unit have already been established in the preceding step, activity design can concentrate solely on the unit's theme and specific needs such as the presentation of knowledge, the forms of student participation, and the training paths of abilities [32], without worrying about global lesson structure. This focus reduces complexity and allows each activity to serve the unit's thematic goals precisely [33,34], avoiding the omission of fine-grained details that often results from trying to balance global coherence.

Through this decomposition and stepwise focusing design logic, the lesson plan is broken down into the three major steps illustrated in Figure 1, taking an example from the Computer Networks course. Step 1 (Teaching Objective Design) converts the course name and session topic into measurable objectives. Step 2 (Instructional Unit Design) partitions the objectives into cognitive-phase-based units across several phases. Step 3 (Teaching Activity Design) generates detailed activities for each unit. The originally complex overall task is thus transformed into a series of sub-tasks with clear goals, well-defined inputs and outputs, and manageable difficulty. Each step focuses on a specific level, progressing from macro-level objectives to micro-level activities. This design ensures the quality of each component while maintaining overall systematic coherence through tight linkage. This decomposition also enables teachers to participate efficiently in the generation process, iteratively improving the lesson plan by controlling the quality of each sub-task.

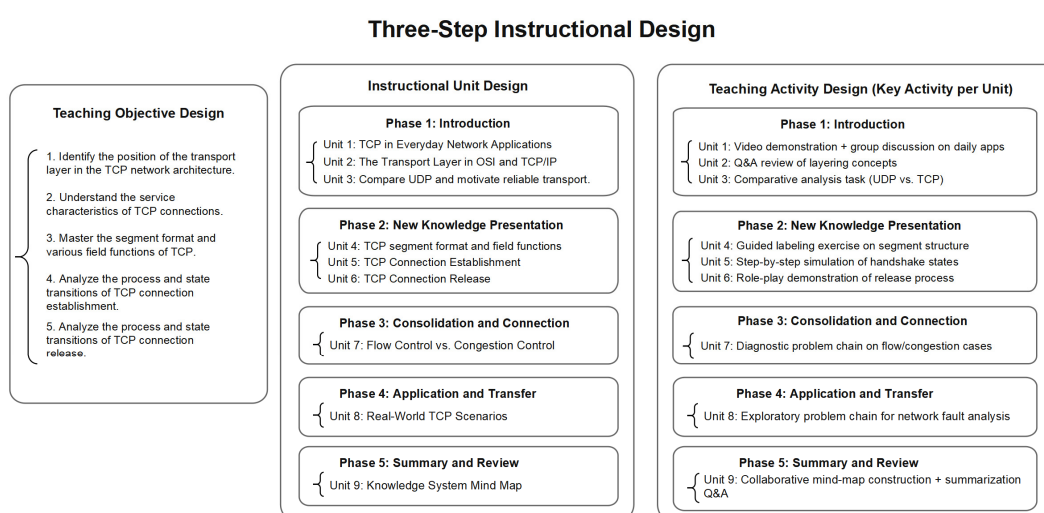


Figure 1. Decomposition of instructional design into three hierarchically nested sub-tasks.

3.2. Three-Step Human–AI Collaborative Lesson Planning Framework

On the overall generative framework, and drawing on the concept of task decomposition, the entire process of lesson plan design is divided into three stages: teaching objective design, instructional unit design, and teaching activity design. Consistent with recent

structured human–AI co-design frameworks [35], this approach preserves the teacher’s leading role as the pedagogical decision-maker and quality gatekeeper [36], realizing a new paradigm of human–AI collaboration that moves beyond the prevailing AI-as-assistant model toward genuine co-design [37]. The top-down, sequentially dependent generation enhances the coherence of the lesson plan through a progressive refinement process, in which each step’s output constrains and informs the next.

For the design of teaching objectives, it references Bloom’s Taxonomy of cognitive levels. For the division of teaching phases, it draws on two alternative cognitive phase models: the three-stage model derived from ACT-R theory and the five-stage model derived from Gagné’s nine events, assigning instructional tasks and cognitive goals to different teaching phases. In terms of teaching activity design, it references the problem-chain-based teaching theory, designing instructional activities based on problem chains, as well as using basic prompts that do not specify a particular theory.

Figure 2 illustrates the complete three-step workflow. In Step 1 (Teaching Objective Design), the model receives the course name and session topic as input and produces a set of measurable learning objectives, guided by Prompt 1 (which incorporates Bloom’s taxonomy). In Step 2 (Instructional Unit Design), the objectives from Step 1 are fed into the model together with Prompt 2 (which incorporates either ACT-R or Gagné’s cognitive phase models), producing a phase-based unit partition. In Step 3 (Teaching Activity Design), the unit plan from Step 2 is combined with Prompt 3 (which optionally includes problem-chain guidance), generating detailed teaching activities for each unit. The output of each step serves as the input for the next, allowing teachers to directly intervene in the generation process of the instructional plan. This step-by-step generation paradigm is reshaping the educational design paradigm of human–AI collaboration. The traditional single-generation mode forced teachers to engage in trial-and-error through vague prompt modifications, whereas phased design deconstructs the creative process into manageable cognitive units.

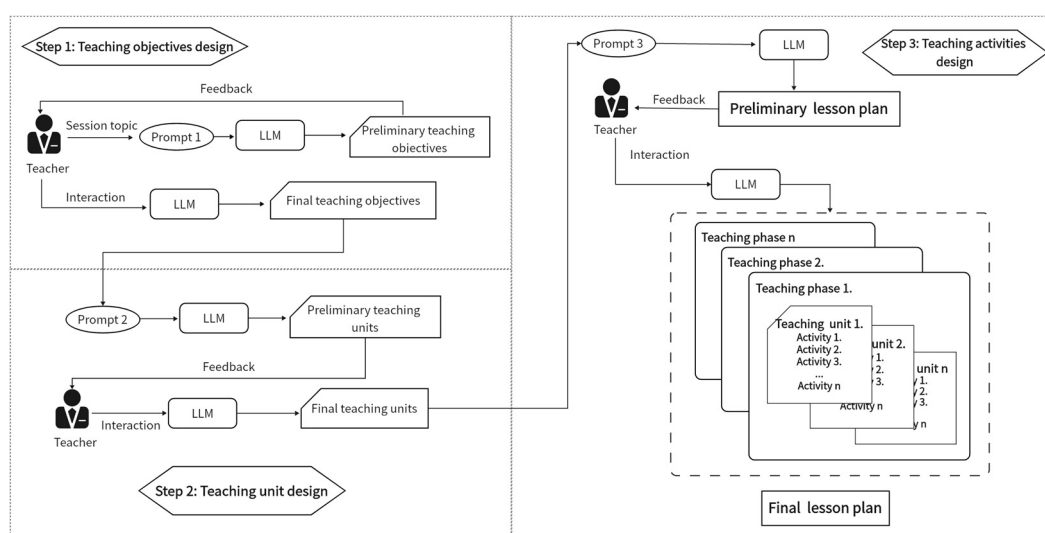


Figure 2. Three-step human–AI collaborative lesson planning framework with corresponding prompts.

3.2.1. Teaching Objective Design Based on Bloom’s Taxonomy

The first step of the framework is teaching objective design. Given the course name and the specific session topic, the model is prompted to generate a set of measurable teaching objectives. This design step is grounded in Bloom’s taxonomy [38,39], a hierarchical framework for classifying educational objectives according to levels of cognitive complexity. The original taxonomy [38] includes six major categories: knowledge, com-

prehension, application, analysis, synthesis, and evaluation. A revised version published in 2001 renamed these categories as remembering, understanding, applying, analyzing, evaluating, and creating, and changed the nouns to verbs to emphasize action [39]. Bloom's taxonomy is widely used by teachers to design teaching objectives that progress from lower-order to higher-order thinking skills. In the context of automated lesson planning, it provides a structured way to formulate measurable, cognitively appropriate teaching objectives, as demonstrated by Hu et al. [6], who used it to generate instructional objectives for GPT-4. By explicitly requiring each objective to pair a specific cognitive level with a concrete knowledge point, the prompt ensures that the generated objectives follow a clear cognitive progression and are stated in observable, assessable terms. To ensure that these objectives follow a clear cognitive progression and are both observable and assessable, the prompt design incorporates Bloom's taxonomy of educational objective.

The prompt template for this step is structured around four key elements. Task background assigns the model the role of an experienced teacher and specifies the discipline and lesson context. Theoretical framework instructs the model to use Bloom's taxonomy, explicitly requiring that each objective combine a cognitive level with a concrete knowledge point. Task description defines the input and the expected output. Output format enforces a structured JavaScript Object Notation (JSON) format and explicitly prohibits any extraneous text, as shown in the Step 1 of Appendix A.2, respectively where {course_name} represents the course title (e.g., "Computer Networks") and {session_topic} represents the specific session topic (e.g., "Transmission Control Protocol (TCP)") within that course.

Using the prompt template, we can obtain the formatted teaching objectives output by the model (see Figure A1 in Appendix B), which follows the template requirements in the prompt, and each one is an executable and verifiable teaching task. These objectives will guide the subsequent step of teaching unit design.

3.2.2. Instructional Unit Design Based on Cognitive Phases

The second step takes the learning objectives generated in Step 1 as input and organizes the lesson into several instructional units, each assigned to a specific cognitive phase. The prompt guides the model to produce a phase-to-unit mapping, and detailed teaching activities will be constructed in Step 3.

As shown in Figure 3, the design of the teaching unit divides the lesson into different teaching stages based on either ACT-R theory or Gagné's nine events. The two serve as parallel alternatives of cognitive-phase models rather than a combination. These two theories were selected because they represent fundamentally different approaches to structuring learning sequences—ACT-R centered on the learner's internal cognitive development, Gagné centered on the teacher's external instructional actions, reflecting a well-established distinction in instructional design theory between learner-centered and instructor-centered paradigms [40]. Comparing them within the same framework allows the study to examine how these divergent pedagogical orientations influence LLM-generated lesson plan quality, providing empirical evidence for RQ2.

1. Three teaching stages based on ACT-R theory

ACT-R (Adaptive Control of Thought-Rational) is a cognitive architecture that explains how humans acquire, represent, and use knowledge [41]. The theory distinguishes between two types of knowledge: declarative knowledge, which consists of facts and concepts stored as chunks, and procedural knowledge, which consists of production rules (if-then condition-action pairs) that guide behavior. Learning is viewed as a transition from declarative to procedural knowledge through practice and feedback. Learning proceeds through three phases [42]: (1) the declarative stage, where learners acquire new facts through explanation and demonstration, forming preliminary conceptual structures; (2) the procedural

stage, where declarative knowledge is converted into production rules through practice and feedback, with errors progressively reduced; and (3) the autonomous stage, where knowledge becomes fully proceduralized, enabling fast, automatic retrieval with minimal cognitive load. This tri-phase model has been validated across mathematics, chemistry, and physics instruction [7].

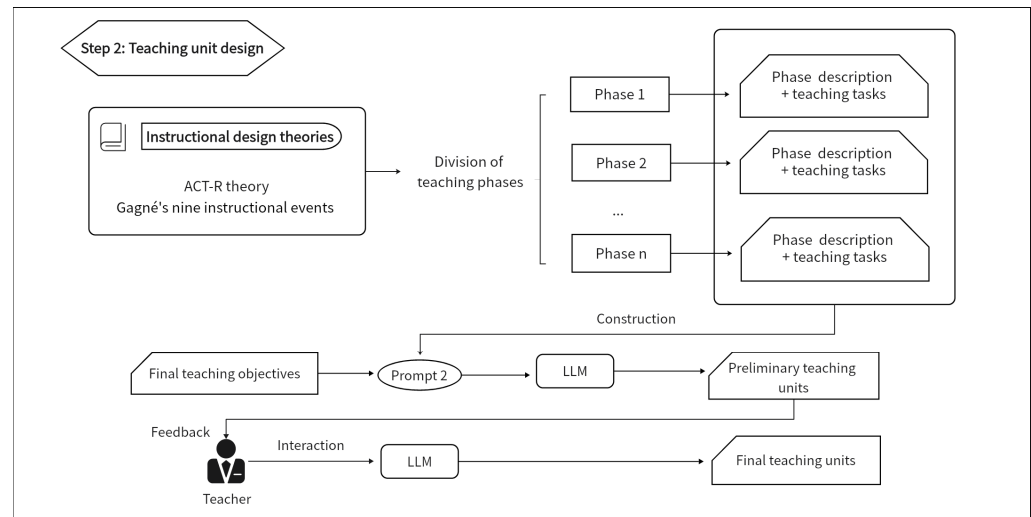


Figure 3. Teaching units design process grounded in pedagogical theories.

In the proposed framework, the model is instructed to allocate instructional units according to these three cognitive stages. ACT-R is particularly appropriate for LLM-generated lesson planning because it provides a clear, cognitively grounded sequence of phases [43] that an LLM can follow to structure content according to the learner's cognitive readiness. The three-stage progression—from declarative knowledge acquisition, through procedural rule formation, to autonomous application—offers a scaffold that compensates for the LLM's lack of intrinsic pedagogical reasoning. By explicitly instructing the model to allocate instructional units to these stages, the prompt guides the LLM toward generating lesson plans that reflect a developmental learning trajectory, potentially resulting in plans that support progressive skill building with clear cognitive differentiation between phases.

2. Five teaching stages based on Gagné's nine events

Robert Gagné's theory of instruction, rooted in information processing psychology, specifies nine events that support internal cognitive processes during learning [32]. These events are: (1) gaining attention, (2) informing learners of the objective, (3) stimulating recall of prior knowledge, (4) presenting the stimulus material, (5) providing learning guidance, (6) eliciting performance, (7) providing feedback, (8) assessing performance, and (9) enhancing retention and transfer. Each event is designed to support a specific internal process, such as selective perception, semantic encoding, or response organization. Gagné's nine events provide a systematic external framework for facilitating the internal stages of human information processing. The sequence moves from capturing attention and setting expectations, through presenting and guiding the processing of new information, to eliciting performance, providing feedback, and finally strengthening retention and transfer. This model highlights the alignment between instructional actions and the learner's cognitive architecture, which have reshaped the design paradigm of modern education and offered a structured basis for designing and evaluating instructional sequences.

Gagné's nine events are appropriate for LLM-generated lesson planning because they translate internal cognitive processes into a concrete sequence of external instructional actions. Each event corresponds to a specific teacher action (e.g., "gain attention," "provide

feedback”) that supports a corresponding cognitive process [44], making this framework highly prescriptive and easy for an LLM to operationalize as a generation guide. By instructing the model to design units aligned with these events, the prompt provides detailed, step-by-step scaffolding that can compensate for the LLM’s lack of teaching experience. This event-based structure is expected to yield lesson plans with clearly defined instructional actions at each phase and strong external coherence, as each event serves a distinct and well-specified pedagogical function.

In authentic classroom practice, however, managing nine discrete events is operationally cumbersome for both teachers and LLM-generated plans. For operational feasibility in classroom practice, this study groups them into five teaching stages following the principle of phase aggregation: (1) Introduction (events 1–3: gaining attention, informing objectives, stimulating recall) prepares cognitive readiness; (2) New Knowledge Presentation (event 4–5: presenting stimulus and providing guidance) handles selective perception of core content; (3) Consolidation and Connection (events 6–7: eliciting performance, providing feedback) forms an integrated practice-and-feedback cycle supporting semantic encoding and response organization; (4) Application and Transfer (events 8–9: assessing performance, enhancing retention and transfer) addresses retrieval and generalization; and (5) Summary and Review, which reconstructs Gagné’s “enhancing retention and transfer” event as a dedicated secondary reinforcement stage. In this stage, structured knowledge networking embeds multiple retrieval cues for long-term memory, while prospective planning and affective closure strengthen long-term anchoring of the learned material. This aggregation preserves each event’s instructional function and is consistent with prior Gagné-based instructional design practice [2,8]. It optimizes operational feasibility for both classroom practice and LLM-generated plans through phase grouping.

The pedagogical differences between ACT-R and Gagné have distinct implications for how the prompt guides the LLM, reflecting a known tradeoff in instructional design between phase granularity and design flexibility [28]. ACT-R’s three-stage model focuses on the learner’s internal cognitive trajectory, providing broader phase boundaries that emphasize the progression of knowledge from declarative to procedural to autonomous. This learner-centric orientation may produce lesson plans that are cognitively coherent and developmentally sequenced, but the broader phase divisions may offer less explicit instructional guidance at the activity level, potentially resulting in plans that lack detailed scaffolding for each sub-phase. Conversely, Gagné’s event-based model focuses on the teacher’s external instructional actions, providing finer phase granularity and more specific pedagogical cues at each step (e.g., distinguishing between “presenting stimulus” and “providing learning guidance”). This teacher-centric, event-based orientation may produce lesson plans with clearer, more differentiated instructional activities and stronger external coherence, but at the cost of potentially greater rigidity.

These theoretical differences are operationalized as two distinct prompting strategies—ACTR-base and Gagné-base, which share the same three-step framework structure but differ in the phase division and instructional cues provided in Step 2. Comparing their outputs within the same experimental design allows the study to examine whether a learner-centric (ACT-R) or a teacher-centric (Gagné) phase structure yields superior lesson plan quality across different evaluation dimensions, thereby offering empirical insights into how divergent pedagogical theories can be operationalized in LLM-generated instructional designs. The prompt template for this step is shown in the Step 2 of Appendices A.2 and A.3. The sample output (ACT-R version used in this example) is illustrated in Figure A2, Appendix B. It can be observed from the output that the generated content strictly adheres to the JSON format. In this stage, based on the overall classroom teaching tasks, the model allocates distinct instructional themes to each classroom teaching phase. This enables the subsequent

design of targeted teaching activities aligned with the pedagogical characteristics of each phase and their corresponding thematic units.

3.2.3. Activity Design Based on Problem-Chain

The third step of the framework takes the instructional unit plan from Step 2 as input and generates detailed teaching activities for each unit. An illustration of this step is presented in Figure 4. Two prompting strategies are designed for the third step, with their detailed configurations described in Section 4.1.2. The first applies no additional pedagogical constraints beyond the phase-level guidance already embedded in the framework, allowing the model to design activities freely. The second enhances the third step with problem-chain theory, explicitly recommending question types for each phase to structure the activity sequence. These two strategies serve as experimental conditions for RQ2, whose comparative effectiveness is examined in the experiment.

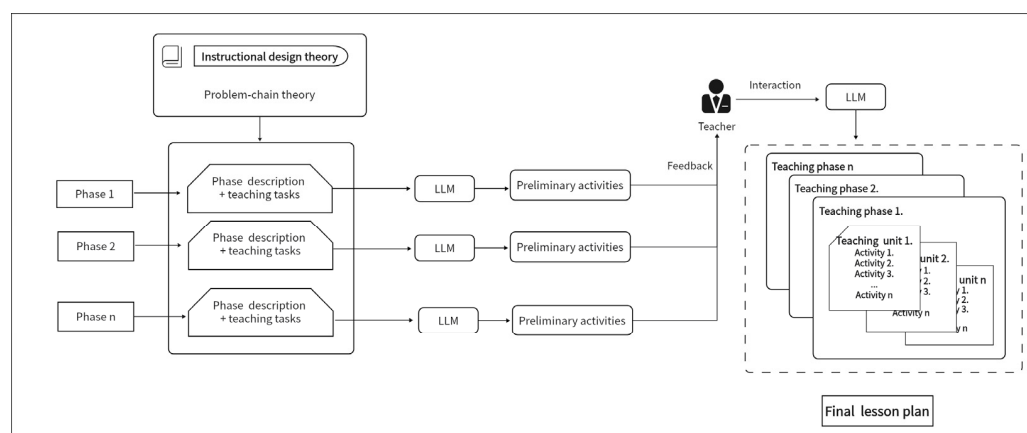


Figure 4. Teaching activity design guided by problem-chain theory.

1. Basic Prompt Design

Similar to the previous steps, the prompt for teaching activity design is organized around four key elements. Task background defines the model's role and the current stage of lesson planning. Theoretical framework describes the cognitive phases of ACT-R or Gagné's five stages and the instructional purpose of each phase. Task description instructs the model to design activities for each unit. Output requirements enforce that each unit receives multiple coherent activities, that the output is structured and parseable, and that no extra commentary is included.

2. Activity Design Based on Problem-Chain

In addition to the basic prompts (ACTR-base and Gagné-base), this study also incorporates problem-chain theory as an additional guiding framework for teaching activity design. The key difference is that the problem-chain-based prompts (ACTR-Probchain and Gagné-Probchain) explicitly include recommended problem-chain types, which effectively reduces the model's task load and improves the quality of the final lesson plan.

Problem-chain teaching theory organizes a lesson around a sequence of interrelated questions rather than direct exposition of facts [45]. A problem chain is a set of questions that share a central theme, are logically connected, and progressively guide students from surface understanding to deep inquiry. Wang [45] categorized problem chains into eight types according to their instructional functions: introductory chains (arouse interest and introduce a topic), diagnostic chains (expose misconceptions), exploratory chains (foster inquiry and creativity), transfer chains (promote knowledge application in new contexts), progressive chains (build complexity step by step), summarization chains (integrate scat-

tered knowledge), and discrepancy chains (create cognitive conflict). In chemistry teaching, Wang demonstrated that diagnostic problem chains increased error identification accuracy from 54% to 82%. Chen [46] applied a three-stage problem chain to geography instruction, showing improved knowledge integration.

In this section, problem-chain theory is integrated into the activity design prompt. The recommended problem-chain types for each teaching phase are derived by mapping the instructional functions of Wang's [45] eight problem-chain categories to the cognitive objectives of each phase. For the ACT-R declarative stage, which emphasizes initial knowledge construction, introductory chains (to introduce topics), transfer chains (to connect to prior knowledge), and difference-based chains (to activate prior knowledge through cognitive conflict) are recommended. The procedural stage, focused on rule formation through practice, is paired with diagnostic, exploratory, and progressive chains. The autonomous stage, targeting automatic application and transfer, is paired with transfer, summarization, and progressive chains. For Gagné's five stages, the same functional logic applies: the introduction and new knowledge presentation stages, which both involve initial exposure to content, receive introductory, transfer, and difference-based chains; the consolidation stage receives transfer, diagnostic, progressive, and difference-based chains to support practice and error correction; the application and transfer stage receives progressive, transfer, and exploratory chains to promote generalization; and the summary stage receives summarization and exploratory chains to support integration and reflection. Table 1 lists the resulting recommended chain types for each phase.

Table 1. Recommended problem-chain types for each teaching phase.

	Teaching Phase	Recommended Problem-Chain Types
ACT-R	Declarative stage	Introductory, transfer, difference-based
	Procedural stage	Diagnostic, exploratory, progressive
	Autonomous stage	Transfer, summarization, progressive
Gagné	Introduction	Introductory, transfer, difference-based
	New knowledge presentation	Introductory, transfer, difference-based
	Consolidation and connection	Transfer, diagnostic, progressive, difference-based
	Application and transfer	Progressive, transfer, exploratory
	Summary and review	Summarization, exploratory

The prompt instructs the model to select one or more chain types for each unit, generate two or more sequenced questions that form a logical chain, and then design concrete teaching activities around those questions. The complete prompt template for the problem-chain guided activity design can be referred to in the Step 3 of Appendices A.4 and A.5, respectively.

A partial output example for a declarative stage unit on "Transmission Control Protocol (TCP)" can be seen in Figure A3, Appendix B. The model first generates a two-question problem chain and then designs four corresponding activities. The resulting activity sequence is coherent, inquiry-driven, and aligned with the declarative stage's goal of building initial understanding.

This problem-chain guided design ensures that the generated activities are not merely descriptive but actively engage students through questioning and reasoning, thereby enhancing the pedagogical depth of the lesson plan. Teachers may edit the generated questions or activities before finalizing the plan.

Based on the preceding design, which involves outlining the overarching instructional tasks, dividing classroom teaching according to cognitive stages, assigning multiple instruc-

tional units to each stage, and specifying the instructional themes for each phase and unit, this stage populates each instructional unit with teaching activities, thereby culminating in the final lesson plan.

4. Methodology

To answer the research questions defined in Section 2.4, the following experiment was conducted.

4.1. Experimental Design

4.1.1. Selection of Large Language Models

Three LLMs of different parameter scales (DeepSeek-R1, DeepSeek-V3 and DeepSeek-R1-Distill-Qwen-32B) were selected to evaluate the generalizability of the proposed framework. DeepSeek is a family of open-weight LLMs developed by DeepSeek-AI, a Chinese AI research company founded in 2023. The company released DeepSeek-V3 in late 2024, with DeepSeek-R1 following in early 2025. The models have gained international recognition for their competitive performance and open accessibility. Specifically, DeepSeek-R1 [47] is a large-scale reasoning model that employs reinforcement learning to enhance chain-of-thought reasoning. On complex reasoning benchmarks, it achieves performance comparable to OpenAI's o1. It was accessed via the official application programming interface. DeepSeek-V3 is a general-purpose model with balanced performance across diverse tasks, comparable to GPT-4o in several benchmarks. It was accessed via the same API. DeepSeek-R1-Distill-Qwen-32B is a distilled variant with 32 billion parameters, based on the Qwen architecture and fine-tuned with DeepSeek-R1 outputs. It represents a compact model suitable for local deployment in resource-constrained environments and was run locally.

These three models were selected for the following reasons. First, they span a wide range of scales and reasoning strengths from the compact 32B distilled model to the full-scale reasoning model, enabling the investigation of model capability as a moderating variable (RQ3). Second, their open-weight nature and public API accessibility enhance the replicability of this study, as other researchers can reproduce the experiments without proprietary access. Third, DeepSeek-R1's strong reasoning capability is particularly relevant to the cognitive structuring demands of lesson planning, allowing us to examine whether stronger reasoning yields deeper pedagogical quality.

4.1.2. Prompting Strategies

Five prompting strategies were designed, each corresponding to a distinct combination of framework structure and pedagogical guidance. Detailed prompt templates are provided in Appendix A.

The Baseline strategy uses a single, unstructured prompt requesting a lesson plan without any pedagogical decomposition or theory guidance. The Gagné-base strategy applies the three-step framework with the five-phase instructional unit design based on Gagné's nine events, but without problem-chain guidance in the activity design step. The ACTR-base strategy applies the three-step framework with the three-phase unit design based on ACT-R theory, also without problem-chain guidance. The Gagné-probchain strategy is identical to Gagné-base except that the activity design step is enhanced with problem-chain theory, including recommended chain types for each phase. Similarly, the ACTR-probchain strategy combines the ACTR-base framework with problem-chain enhanced activity design. These five strategies enable a systematic comparison of the contribution between the Baseline and theory-based framework and the additional contribution of problem-chain guidance.

4.1.3. Lesson Planning Procedure

Ten lessons were extracted from the course syllabus of the Computer Networks course on the Chinese MOOC platform as shown in Table 2. Specifically, ten session topics were sampled from the syllabus to ensure that the input themes are authentic, professionally designed, and representative of a standard university course. For each topic, lesson plans were generated independently using each of the three models and each of the five prompting strategies, resulting in 15 plans per topic and a total corpus of 150 lesson plans. All generations were performed with a fixed temperature setting of 0.7 and a fixed random seed to ensure reproducibility. The temperature parameter controls the randomness of text generation by scaling the probability distribution over candidate tokens before sampling [48]; a lower value (near 0) makes outputs more deterministic, while a higher value (near 1) produces greater diversity. The value 0.7 is a widely adopted default that balances coherence with variation [19]. The fixed random seed initializes the pseudo-random number generator to a consistent starting state, so that identical inputs and parameters produce identical outputs, which is a standard practice for reproducibility in LLM-based research [15,19].

Table 2. Ten session topics of Computer Networks on Chinese MOOC platform.

No.	Session Topic
1	Basic Concepts of Computer Networks
2	Computer Network Architecture
3	Network Core
4	Computer Network Performance
5	Computer Network Architecture
6	Overview of Network Application (Layer) Content
7	Basic Principles of Network Applications
8	Web Application
9	Email Application
10	Transmission Control Protocol (TCP)

Each plan was saved as a plain text file with metadata for later blind evaluation. This design produces a balanced corpus that allows within-topic, cross-model, and cross-strategy comparisons.

4.2. Quality Evaluation Methods

Current evaluation approaches for open-ended educational texts such as lesson plans can be broadly categorized into two types: indicator-based assessment and outcome-based evaluation. Indicator-based approaches evaluate the internal quality of the plan itself against a predefined rubric, using either human experts or LLMs as judges. For instance, Clark et al. [49] employed an LLM-as-Judge method with quality benchmarks to enable batch scoring, while Zheng et al. [8] asked three experienced teachers to assess whether each plan contained Gagné’s nine instructional events. Outcome-based approaches, in contrast, validate lesson plan quality through downstream effects on student performance or learning gains [4]. Despite their greater ecological validity, outcome-based methods face significant practical barriers, including long instructional cycles and high resource demands. This study therefore adopts the indicator-based approach, combining automated LLM-based evaluation with human validation on a stratified sub-sample. Two comple-

mentary evaluation dimensions are employed across both tracks: coverage of Gagné’s nine instructional events and functional quality based on a 12-indicator scoring rubric.

4.2.1. Automated Evaluation by LLM Judge

1. Coverage of Gagné’s nine instructional events

Gagné’s nine events provide a well-established framework for describing a complete instructional sequence. Each event supports a corresponding internal cognitive process, making this framework suitable for evaluating structural completeness. The functional quality 12-indicator rubric and blind human ratings discussed in following sections serve as two complementary safeguards that address the concern that Gagné-based prompts may be inherently advantaged by this choice of evaluation criteria.

The DeepSeek-R1 API was used to make a binary determination of whether a given lesson plan explicitly includes each of the nine events. The evaluation prompt specified: (1) a clear definition and description of each event, (2) an explicit instruction: “Please determine whether the given teaching plan explicitly includes the XXX teaching event”, and (3) a structured JSON output format. Multiple pre-tests confirmed that the model’s outputs were 100% parseable. The event coverage score for each plan was computed as the proportion of the nine events judged as present.

2. Functional quality based on a multi-indicator scoring rubric

Beyond the presence or absence of instructional events, pedagogical quality requires a more nuanced assessment. Following the LLM-as-Judge approach [49], the same DeepSeek-R1 model was used to assign functional quality scores via its official API [47]. A three-level discrete scale (1 = poor, 3 = acceptable, 5 = excellent) was adopted instead of a continuous percentage scale, because discrete grading better aligns with the model’s semantic understanding of quality descriptors and avoids instability caused by fine-grained numerical mapping. Although the model’s conceptualization of “excellence” may diverge from that of instructional experts, the alignment with human preferences via reinforcement learning exemplified by the GRPO technique employed in DeepSeek [47], renders its three-tier evaluation framework highly instructive.

A scoring rubric was developed comprising twelve indicators grouped into three dimensions (Table 3): teaching objectives (A1–A4), teaching activities (B5–B8), and content knowledge (C9–C12). For each indicator, the judge was prompted to provide both a score and a brief justification. This requirement explicitly activates the model’s chain-of-thought reasoning mechanism [50], which has been shown to improve accuracy and reliability in evaluation tasks by making the reasoning process explicit and traceable. The judge’s prompt defined each indicator clearly and instructed the model to output a structured JSON object containing the score and the corresponding justification for each indicator. Preliminary validation confirmed that the judge’s outputs were fully parsable and consistent.

Table 3. Twelve indicators for functional quality scoring, organised by dimension.

Dimension	Indicator	Description
Teaching objectives	A1. Alignment with core literacy	Objectives reflect subject-specific core competencies
	A2. Cognitive level structuring	Objectives follow Bloom’s taxonomy levels
	A3. Focus on key content	Objectives target essential knowledge points
	A4. Measurability	Objectives contain observable action verbs

Table 3. *Cont.*

Dimension	Indicator	Description
Teaching activities	B5. Alignment with objectives	Activities directly serve the stated objectives
	B6. Cognitive progression	Activities follow a perception → understanding → application → creation sequence
	B7. Student engagement	Activities involve discussion, presentation, or hands-on tasks
	B8. Multi-sensory design	Activities integrate visual, auditory, and kinesthetic channels
Content knowledge	C9. Knowledge structure	Knowledge points are sequenced according to disciplinary logic
	C10. Interdisciplinary integration	Cross-disciplinary content is included and serves core goals
	C11. Completeness of core knowledge	All key knowledge points of the topic are covered
	C12. Design for difficult points	Special activities are designed for high-complexity content

4.2.2. Human Validation of Automated Evaluation

To validate the LLM-based automated evaluation and address the potential circularity concern arising from using a model from the same family as both generator and judge, a human evaluation study was conducted. The human ratings served a confirmatory validation role rather than a primary evidentiary one. The main conclusions are drawn from the full-corpus LLM evaluations ($n = 150$), with human ratings providing convergent evidence to assess the reliability of the LLM judge.

A stratified random sample of 30 lesson plans was drawn from the full corpus of 150 plans, covering all 5 prompting strategies and 3 model variants with 2 distinct topics per cell. This sampling strategy ($5 \text{ strategies} \times 3 \text{ models} \times 2 \text{ topics} = 30 \text{ plans}$) ensured balanced representation of all experimental conditions while maintaining a feasible rating workload for human experts. Each plan was anonymized and assigned a random identifier (P01–P30) so that raters were blind to the strategy and model used to generate each plan.

Five raters ($M_{\text{age}} = 39.4$, $SD = 4.7$; years of teaching experience: $M = 6.2$, $SD = 3.1$; 3 male, 2 female) specializing in computer science were recruited from university. All raters held at least a doctoral degree, had at least 2 years of experience in course design and lesson plan development, and were familiar with the Computer Networks curriculum. None had prior exposure to the generated lesson plans or the study's experimental design.

Each rater independently judged whether the plan contained each of Gagné's nine instructional events using the same binary determination protocol as the LLM judge. The event coverage score was computed as the proportion of the nine events judged as present. To improve discriminability compared to the 3-level scale used in the automated evaluation, a 5-level behavioral anchoring scale was adopted (1 = Poor, 2 = Basic, 3 = Acceptable, 4 = Good, 5 = Excellent), with specific behavioral descriptors defined for each indicator level. The mean score across all 12 indicators was used as the overall functional quality score per plan. The different scale granularity reflects the complementary roles of the two evaluation tracks: the LLM judge prioritizes reliability and consistency through a small set of well-separated anchors, while human raters leverage the finer scale to capture nuanced quality distinctions that automated evaluation may overlook.

Prior to formal rating, raters completed a 60 min training session including: (1) a briefing on the study's purpose and the 2 evaluation methods, (2) independent scoring of 2 practice plans, and (3) group discussion of scoring discrepancies to calibrate their

understanding of the criteria. Following training, each rater independently scored all 30 plans using the same 12-indicator rubric (Table 3).

Inter-rater reliability was assessed using the intraclass correlation coefficient ICC(2,k) under a two-way random-effects model for absolute agreement of average measures [51]. Concurrent validity between human ratings and LLM judge ratings was assessed using Pearson and Spearman correlations. Systematic bias was quantified as the mean difference between human mean scores and LLM judge scores. All statistical analyses were conducted in Python 3.10.12 using the pingouin library (version 0.6.1) [52].

5. Results

This section reports evaluation results across two data tracks: the full-corpus automated evaluation using the LLM judge on all 150 lesson plans, and the human validation on a stratified sub-sample of 30 plans rated by five expert teachers. For each research question, the full-corpus automated data are presented first, followed by the corresponding human validation data. Per-indicator analyses (RQ2) are based solely on the full-corpus LLM data, as the human sub-sample lacks sufficient cell sizes for 12-indicator and 5-strategy decomposition.

5.1. Systematic Enhancement of Instructional Event Coverage and Functional Quality

5.1.1. Instructional Event Coverage

For full-corpus automated evaluation ($n = 150$), Table 4 presents the average coverage of Gagné's nine events across all five strategies and three models. All theory-based strategies achieved coverage above the Baseline for every model. The largest improvement was observed for the 32B model, where ACTR-probchain raised coverage from 71.6% to 98.7%. For DeepSeek-R1, the highest coverage was achieved by Gagné-base (98.8%).

Table 4. Average coverage (%) of Gagné's nine instructional events (LLM judge, $n = 150$).

Model	Baseline	Gagné-Base	ACTR-Base	Gagné-Probchain	ACTR-Probchain
DeepSeek-R1	86.43	98.76	95.06	96.30	93.83
DeepSeek-V3	90.13	95.06	97.53	96.30	98.76
32B-Distill	71.60	83.94	92.58	87.66	98.70

Table 5 presents the human-judged event coverage on the 30-plan sub-sample alongside the LLM scores for the same plans. Inter-rater reliability was ICC(2,k) = 0.786, 95% CI [0.621, 0.893]. The rank ordering of strategies was consistent between the two evaluation sources, with all theory-based strategies exceeding 89% in human ratings compared to 74.1% for Baseline. The LLM judge scored every strategy higher than the human raters, with the gap ranging from 3.7 to 8.3 percentage points. Despite this upward bias, the rank ordering of strategies was preserved between the two evaluation sources. The Pearson correlation between human-judged and LLM-judged event coverage at the plan level was $r = 0.724$, $p < 0.001$; Spearman's $\rho = 0.746$, $p < 0.001$.

5.1.2. Functional Quality Scores

In terms of full-corpus automated evaluation ($n = 150$), Table 6 presents the mean functional quality scores (1–3 scale, 1 = Poor, 3 = Acceptable, 5 = Excellent) across all conditions. The strongest reasoning model DeepSeek-R1 achieved the highest score under ACTR-probchain (4.82). The 32B model showed the largest improvement from Baseline to the best strategy (3.92 to 4.52).

Table 5. Event coverage (%) on the human validation sub-sample ($n = 30$).

Strategy	Human Mean (SD)	LLM Mean (SD)
Baseline	74.1 (8.2)	82.4 (6.7)
Gagné-base	92.6 (4.1)	96.3 (3.9)
ACTR-base	89.8 (5.6)	94.4 (4.8)
Gagné-probchain	91.7 (4.9)	96.3 (3.4)
ACTR-probchain	93.5 (3.7)	97.2 (3.1)

Note. $N = 6$ plans per strategy. Human scores are the mean across five raters.

Table 6. Mean functional quality scores (LLM judge).

Model	Baseline	Gagné-Base	ACTR-Base	Gagné-Probchain	ACTR-Probchain
DeepSeek-R1	4.11	4.78	4.51	4.78	4.82
DeepSeek-V3	4.25	4.66	4.19	4.7	4.79
32B-Distill	3.92	4.35	4.35	4.52	4.37

Table 7 presents the human-rated functional quality scores on the 30-plan sub-sample. Inter-rater reliability was $ICC(2,k) = 0.811$, 95% CI [0.657, 0.905]. The Pearson correlation between human and LLM scores was $r = 0.738$, $p < 0.001$; Spearman's $\rho = 0.757$, $p < 0.001$. The LLM judge assigned higher scores on 25 of 30 plans, with a mean upward bias of 0.43 points (SD = 0.37). The rank ordering of strategies was preserved across evaluation sources: Gagné-base received the highest human rating ($M = 4.11$), followed by ACTR-probchain ($M = 4.09$), while Baseline was the lowest ($M = 3.13$).

Table 7. Functional quality scores on the human validation sub-sample ($n = 30$, 1–5 scale).

Strategy	Human Mean (SD)	LLM Mean (SD)	LLM-Human Difference
Baseline	3.13 (0.33)	3.65 (0.65)	+0.52
Gagné-base	4.11 (0.25)	4.35 (0.37)	+0.24
ACTR-base	3.87 (0.41)	4.18 (0.45)	+0.31
Gagné-probchain	4.02 (0.15)	4.63 (0.27)	+0.61
ACTR-probchain	4.09 (0.46)	4.58 (0.30)	+0.49

Note. $N = 6$ plans per strategy. Human scores are the mean across five raters.

The relationship between the two evaluation sources is shown in Figure 5. The majority of data points fall above the identity line, reflecting the upward bias reported in Table 7. Baseline plans cluster in the lower-left region, and the OLS regression slope (0.71) indicates that the LLM judge differentiated quality in the same direction as human raters, albeit with compressed sensitivity at the high end.

5.1.3. Cross-Metric Summary

Figure 6 summarizes the strategy-level comparison across both evaluation metrics. In panel (a), human-rated event coverage ranged from 74.1% (Baseline) to 93.5% (ACTR-probchain), while LLM-rated coverage ranged from 82.4% to 97.2%. In panel (b), human-rated functional quality ranged from 3.13 to 4.11, whereas LLM-rated scores ranged from 3.65 to 4.63. Across both panels, the rank ordering of strategies was consistent between human and LLM evaluation, with all theory-based strategies scoring above Baseline. The LLM judge assigned higher scores for every strategy on both metrics, and the score range was wider in human ratings than in LLM ratings, particularly for functional quality.

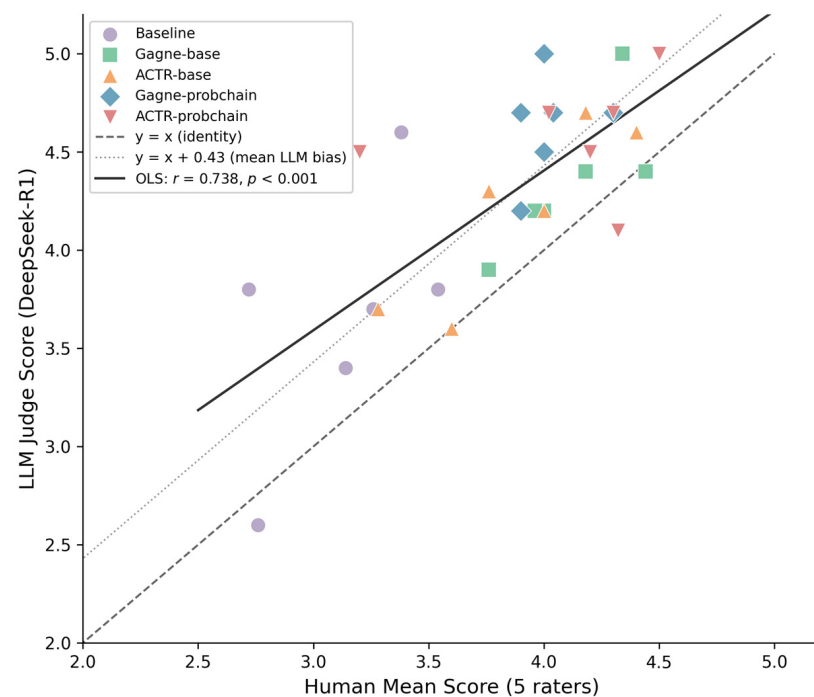


Figure 5. Human mean functional quality scores vs. LLM judge scores for the 30-plan validation sub-sample.

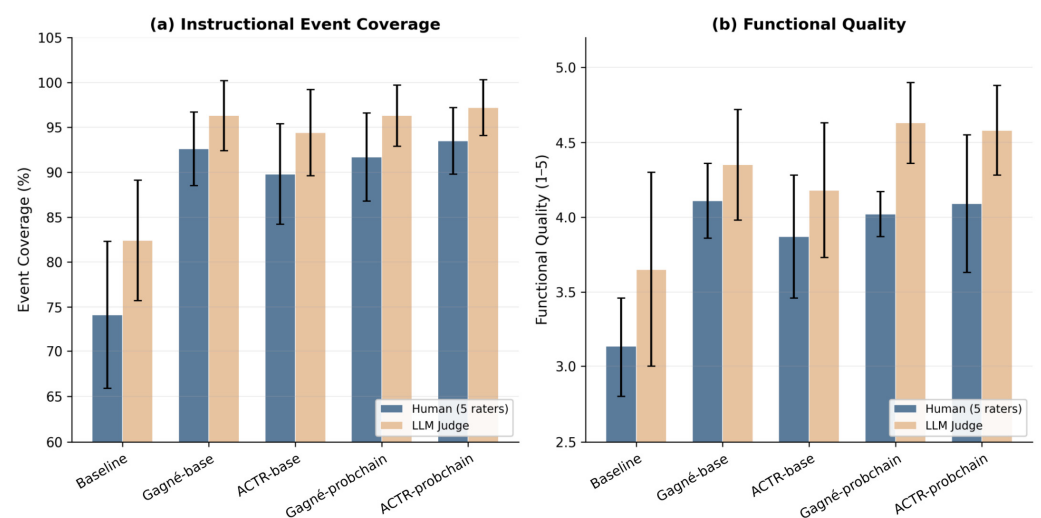


Figure 6. Strategy-level comparison of human and LLM evaluation on the 30-plan validation sub-sample.

5.2. Contribution of Cognitive Phase Models and Problem-Chain Guidance

The following per-indicator analyses are based on the full-corpus LLM evaluation ($n = 150$) using DeepSeek-R1. The human validation sub-sample provides strategy-level convergent evidence only.

5.2.1. Comparison of Cognitive Phase Models

Table 8 reports the 12-indicator functional quality scores for ACTR-base and Gagné-base using DeepSeek-R1. Gagné-base scored higher than ACTR-base on A1 (+0.66), A2 (+1.00), and C10 (+1.55). Indicators A3, A4, B5–B8, C9, C11, and C12 received the maximum score for both strategies. The overall mean was 4.78 for Gagné-base and 4.51 for ACTR-base.

Table 8. Functional quality scores of ACTR-base and Gagné-base for DeepSeek-R1 (LLM judge).

Indicator	ACTR-Base	Gagné-Base	Difference
A1 (core literacy alignment)	3.67	4.33	+0.66
A2 (cognitive level structuring)	2.89	3.89	+1.00
A3 (focus on key content)	5.00	5.00	0
A4 (measurability)	5.00	5.00	0
B5 (activity-objective alignment)	5.00	5.00	0
B6 (cognitive progression)	5.00	5.00	0
B7 (student engagement)	5.00	5.00	0
B8 (multi-sensory design)	5.00	5.00	0
C9 (knowledge structure)	5.00	5.00	0
C10 (interdisciplinary integration)	2.56	4.11	+1.55
C11 (core knowledge completeness)	5.00	5.00	0
C12 (difficult-point design)	5.00	5.00	0
Overall mean	4.51	4.78	+0.27

At the aggregated strategy level on the human validation sub-sample, Gagné-base ($M = 4.11$) also scored higher than ACTR-base ($M = 3.87$), as shown in Table 7.

5.2.2. Additional Effect of Problem-Chain Guidance

Table 9 presents the indicator-level comparison of ACTR-probchain vs. ACTR-base and Gagné-probchain vs. Gagné-base. For the ACT-R framework, the addition of problem-chain guidance increased the overall mean from 4.51 to 4.82, with the largest gains in A2 (+1.55) and A1 (+1.33). For the Gagné framework, the overall mean remained unchanged at 4.78.

Table 9. Effect of problem-chain guidance on 12-indicator scores for DeepSeek-R1 (LLM judge).

Indicator	ACTR-Base	ACTR-Probchain	Difference	Gagné-Base	Gagné-Probchain	Difference
A1	3.67	5.00	+1.33	4.33	4.44	+0.11
A2	2.89	4.44	+1.55	3.89	3.78	−0.11
A3	5.00	5.00	0	5.00	5.00	0
A4	5.00	5.00	0	5.00	5.00	0
B5	5.00	5.00	0	5.00	5.00	0
B6	5.00	5.00	0	5.00	5.00	0
B7	5.00	5.00	0	5.00	5.00	0
B8	5.00	5.00	0	5.00	5.00	0
C9	5.00	5.00	0	5.00	5.00	0
C10	2.56	3.33	+0.77	4.11	4.11	0
C11	5.00	5.00	0	5.00	5.00	0
C12	5.00	5.00	0	5.00	5.00	0
Overall mean	4.51	4.82	+0.31	4.78	4.78	0

On the human-rated sub-sample (Table 7), ACTR-probchain ($M = 4.09$) exceeded ACTR-base ($M = 3.87$) by 0.22 points, while Gagné-probchain ($M = 4.02$) and Gagné-base ($M = 4.11$) differed by only 0.09 points in the opposite direction. These magnitudes are consistent with the full-corpus pattern in which problem-chain guidance substantially improved the ACT-R framework but had negligible effect on the Gagné framework.

5.3. Performance Across Different Model Scales

To examine how model capability moderates the effectiveness of the proposed framework, three models of different scales and reasoning strengths are compared: DeepSeek-R1 with strong reasoning, DeepSeek-V3 with moderate reasoning, and DeepSeek-R1-Distill-Qwen-32B with weaker reasoning. Two metrics are used for this comparison: the average coverage of Gagné's nine instructional events and the mean functional quality score based on the 12-indicator rubric. To ensure comparability, the same prompting strategy, ACTR-probchain, is applied to all three models in both metrics. ACTR-probchain is selected because it achieves the highest or near-highest scores for all models across both metrics.

5.3.1. Instructional Event Coverage by Model

Table 10 presents the full-corpus event coverage by model under Baseline and ACTR-probchain. The 32B model showed the largest improvement (from 71.6% to 98.7%, +27.1 pp), while DeepSeek-R1 and DeepSeek-V3 showed smaller relative gains (+7.4 and +8.63 pp, respectively).

Table 10. Average coverage of Gagné's nine events across models (LLM judge).

Model	Baseline (%)	ACTR-Probchain (%)	Difference (pp)
DeepSeek-R1	86.43	93.83	+7.40
DeepSeek-V3	90.13	98.76	+8.63
32B-Distill	71.6	98.7	+21.10

On the 30-plan sub-sample, as illustrated in Table 11, the pattern of model-scale moderation was consistent: the 32B model showed the largest human-rated event coverage gain from Baseline (68.5%) to ACTR-probchain (94.4%, +25.9 pp), compared to gains of +13.0 and +20.3 pp for DeepSeek-R1 and DeepSeek-V3, respectively.

Table 11. Event coverage (%) by model on the human validation sub-sample ($n = 30$).

Model	Baseline Mean (SD)	ACTR-Probchain Mean (SD)	Difference (pp)
DeepSeek-R1	78.7 (7.4)	91.7 (3.6)	+13.0
DeepSeek-V3	74.1 (6.2)	94.4 (3.1)	+20.3
32B-Distill	68.5 (9.1)	94.4 (2.8)	+25.9

Note. $N = 2$ plans per model per strategy. Scores are the mean across five raters and two plans.

5.3.2. Functional Quality Scores by Model

Table 12 reports the full-corpus functional quality scores by model. DeepSeek-R1 achieved both the highest absolute score (4.82) and the largest gain (+0.71 points). DeepSeek-V3 showed a gain of 0.54 points. The 32B model, despite starting from the lowest Baseline (3.92), showed the smallest gain (+0.45 points) and the lowest final score (4.37). This pattern, in which the strongest reasoning model benefited most from the framework in both absolute and relative terms, differs from the moderation pattern observed for event coverage (See Section 5.3.1), where smaller models benefited more.

Table 12. Mean functional quality scores across models (LLM judge).

Model	Baseline	ACTR-Probchain	Difference (Points)
DeepSeek-R1	4.11	4.82	+0.71
DeepSeek-V3	4.25	4.79	+0.54
32B-Distill	3.92	4.37	+0.45

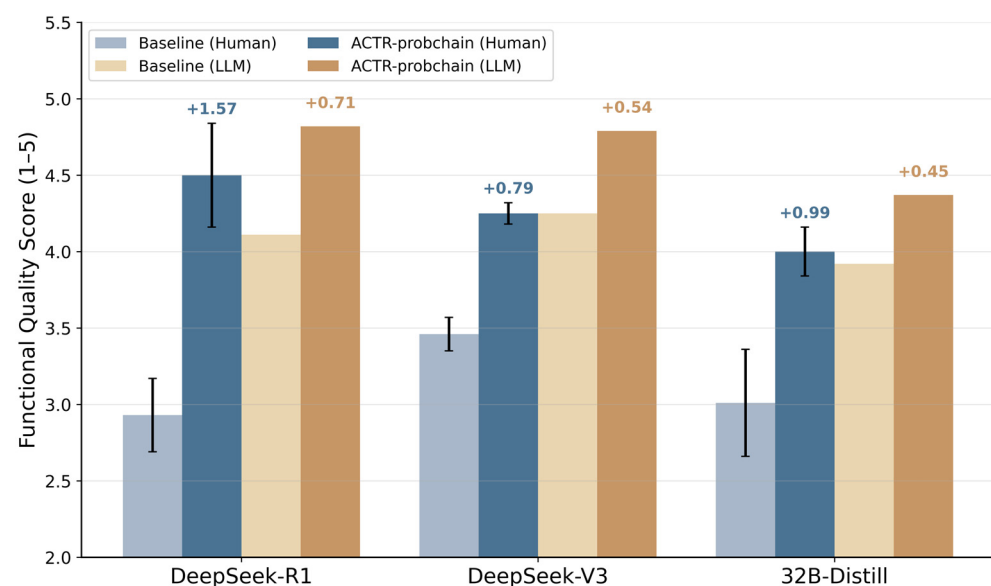
Table 13 presents the human-rated functional quality scores by model. DeepSeek-R1 again showed the largest gain (+1.57 points), while DeepSeek-V3 showed the smallest gain (+0.79 points). The 32B model fell between the two larger models in the difference (+0.99 points) but remained the lowest in absolute terms (4.00). The convergence of the full-corpus and human-validation results on the same direction that DeepSeek-R1 benefits most in both absolute and relative functional quality gains.

Table 13. Functional quality scores by model on the human validation sub-sample ($n = 30$, 1–5 scale).

Model	Baseline Mean (SD)	ACTR-Probchain Mean (SD)	Difference (Points)
DeepSeek-R1	2.93 (0.24)	4.50 (0.34)	+1.57
DeepSeek-V3	3.46 (0.11)	4.25 (0.07)	+0.79
32B-Distill	3.01 (0.35)	4.00 (0.16)	+0.99

Note. $N = 2$ plans per model per strategy. Scores are the mean across five raters and two plans.

Figure 7 summarizes the model-scale moderation effect. In both evaluation tracks, DeepSeek-R1 attained the highest absolute score and the largest gain under ACTR-probchain, while the 32B model gained proportionally more than DeepSeek-V3 in human ratings but remained the lowest in absolute terms.

**Figure 7.** Functional quality scores by model under Baseline and ACTR-probchain prompting.

6. Discussion

6.1. Overall Effectiveness of the Prompting Framework (RQ1)

The results confirm that the proposed three-step pedagogy-driven prompt framework systematically enhances both the completeness and the quality of LLM-generated lesson plans. In the full corpus, all theory-based strategies raised the average coverage of Gagné's

nine events above 90%. The functional quality scores based on the 12-indicator rubric also improved substantially under theory-based strategies relative to the baseline. In the human sub-sample, theory-based strategies achieved 89.8% to 93.5% coverage versus 74.1% for Baseline, and functional quality scores of 3.87 to 4.11 versus 3.13 for Baseline. The convergence of human and LLM evaluations on strategy ranking ($r = 0.724$, Spearman's $\rho = 0.746$ for event coverage; $r = 0.738$, Spearman's $\rho = 0.757$ for functional quality) lends credibility to the principal finding. These findings indicate that decomposing lesson planning into three structured sub-tasks, namely objective design, unit design, and activity design, and grounding each sub-task in established pedagogical theories effectively compensates for the lack of intrinsic pedagogical reasoning in LLMs [53]. This pattern echoes findings by Zhang et al. [54] in LLM-based instructional design, where decomposing lesson planning across multiple specialized LLM agents consistently outperformed monolithic generation, supporting the principle that finer task granularity yields higher-quality plans.

The human validation also clarifies the magnitude of improvement. Whereas the full-corpus LLM scores placed most strategies near the scale ceiling (above 4.5 points), human ratings distributed them across a wider range (3.13 to 4.11), suggesting that the framework's absolute gains, while substantial, are more modest than LLM-only data would imply. The consistent upward bias of the LLM judge (+0.43) is partially attributable to the leniency tendencies in LLM-based evaluation [55] and different scale granularity—the LLM used a three-level scale (1, 3, 5) whereas human raters used a five-level scale (1–5) with an intermediate option of 4. The full-corpus scores should therefore be interpreted as upper-bound estimates rather than as ground-truth judgments of absolute quality.

6.2. Contribution of Cognitive Phase Models and Problem-Chain Guidance (RQ2)

The comparison between ACT-R and Gagné cognitive phase models revealed that Gagné's five-stage design outperformed ACT-R's three-stage design under the base condition. This advantage was most evident in cognitive level structuring (A2) and interdisciplinary integration (C10), suggesting that a teacher-centric, event-based structure (Gagné) yields stronger performance on these dimensions than a learner-centric, cognitively oriented structure (ACT-R) when no additional pedagogical scaffolding is provided. A plausible explanation, grounded in cognitive load theory, is that Gagné's finer phase division reduces interpretive ambiguity for the model by providing more explicit pedagogical cues at each step [56]. For example, distinguishing between “presenting stimulus material” and “providing learning guidance” helps the model allocate appropriate instructional actions more precisely. In contrast, ACT-R's broader declarative, procedural, and autonomous stages may leave more room for interpretation, making it harder for the model to generate cognitively differentiated activities without additional guidance.

The addition of problem-chain guidance produced substantial gains for the ACT-R framework but had negligible effect on the Gagné framework. This asymmetry was replicated in human ratings, where ACTR-probchain exceeded ACTR-base by 0.22 points while Gagné-probchain differed from Gagné-base by only 0.09 points. These convergent findings suggest that problem-chain theory functions as a compensatory scaffolding mechanism [57] compensating for the coarser phase structure of ACT-R. When the model is given only three broad stages, explicit instructions to use introductory, transfer, or difference-based problem chains [45] help break down each stage into a sequence of inquiry-driven steps, thereby improving cognitive level structuring and interdisciplinary integration. For Gagné, the five stages already embed a detailed instructional sequence, leaving little room for further refinement from problem-chain prompts.

A methodological observation warrants mention: in the full-corpus LLM evaluation, eight of twelve indicators received maximum scores across all theory-based strategies. This

ceiling effect limits the rubric's discriminative power at the per-indicator level. Distinct from the 3-level scale used for automated evaluation, the human validation partially addressed this limitation through a finer-grained 5-level behavioral anchoring scale, under which even the best-performing strategy (Gagné-base, $M = 4.11$) remained below the scale maximum on most indicators.

6.3. Moderation by Model Scale and Reasoning Capability (RQ3)

The moderation pattern differed between the two evaluation dimensions. For instructional event coverage, smaller models benefited most: the 32B model showed the largest gain in both the full corpus (+27.1 pp) and the human sub-sample (+25.9 pp). This is attributable to the fact that structural completeness has a reachable ceiling: larger models already embed a relatively complete pedagogical sequence, whereas smaller models require explicit structural prompts to achieve comparable coverage.

For functional quality, the pattern reversed: DeepSeek-R1 achieved both the largest gain (+0.72 full corpus, +1.57 human) and the highest absolute score (4.82 full corpus, 4.50 human). This suggests that functional quality, unlike structural completeness, draws on deeper pedagogical reasoning that stronger models are better equipped to leverage from theory-grounded prompts. This aligns with documented emergent phenomena in LLMs, where complex reasoning capabilities manifest only above certain scale thresholds [58]. The 32B model, despite showing a notable gain in human ratings (+0.99 points), remained the lowest in absolute terms (4.00), confirming a capacity ceiling for smaller models on cognitively demanding evaluation dimensions [15].

The human validation revealed an additional nuance: the gap between Baseline and ACTR-probchain was consistently larger in human ratings than in LLM ratings for all three models (Figure 7). This difference suggests that human raters are not only more conservative in absolute scoring but also more sensitive to the qualitative improvements introduced by the framework [59].

The overall picture is one of complementary strengths: the framework's structured prompts provide the largest structural boost to resource-constrained models. Meanwhile, stronger reasoning models extract deeper pedagogical value from the same prompts, offering a higher quality ceiling for institutions with access to more capable LLMs.

7. Conclusions

7.1. Summary of Research

This study proposed a three-step pedagogy-driven prompt engineering framework to address the limitations of naive prompting in LLM-generated lesson planning. The framework decomposes lesson planning into three structured sub-tasks: learning objective design, instructional unit design, and teaching activity design. Each sub-task is grounded in established pedagogical theories (Bloom's taxonomy, ACT-R cognitive theory, Gagné's nine instructional events, problem-chain theory), and allows teachers to intervene at any intermediate step, enabling fine-grained human-AI collaboration.

Three LLMs (DeepSeek-R1, DeepSeek-V3, and DeepSeek-R1-Distill-Qwen-32B), and five prompting strategies were evaluated using ten session topics from the Computer Networks course on Chinese MOOC platform. Two complementary metrics were used: coverage of Gagné's nine instructional events and functional quality scores based on a 12-indicator rubric. Evaluation was conducted across two tracks: a full-corpus automated evaluation using an LLM judge ($n = 150$) and a human validation on a stratified sub-sample rated by five expert teachers ($n = 30$).

For RQ1 (framework effectiveness), all theory-based strategies outperformed naive prompting in both evaluation tracks. Average instructional event coverage exceeded 90%

across all models in the full corpus, with human-rated coverage showed the same pattern. Functional quality scores improved in both tracks, with human and LLM scores showing strong convergent validity. The LLM judge exhibited a systematic upward bias, indicating that the full-corpus scores should be interpreted as upper-bound estimates. For RQ2 (pedagogical contributions), Gagné's five-stage cognitive phase design outperformed ACT-R's three-stage design under base conditions, and problem-chain guidance provided substantial gains for the ACT-R framework but had little effect on Gagné. These patterns were corroborated in the human sub-sample at the strategy level. For RQ3 (model moderation), the moderation pattern differed by dimension. In instructional event coverage, the smallest model 32B benefited most. In functional quality, DeepSeek-R1 achieved both the largest relative gain and the highest absolute score in both evaluation tracks, while the 32B model remained the lowest in absolute terms despite notable improvement. This suggests that structural completeness is more readily compensated by explicit prompts, whereas functional depth draws on reasoning capabilities that scale with model capacity.

7.2. Theoretical Implications

This study contributes to the literature on educational technology and prompt engineering in several ways. Pedagogical theories can be operationalized as structured prompt components that guide LLM behavior in a principled manner. By decomposing lesson planning into three sub-tasks and explicitly mapping each to a specific pedagogical theory, the framework provides a replicable template for embedding domain knowledge into LLM prompts. This bridges the gap between abstract educational principles and machine-executable instructions.

The multi-stage intervention model offers a conceptual foundation for designing more flexible, teacher-centered AI systems where prompt engineering should be understood not merely as a narrow technical skill, but as a form of professional practice for teachers [60]. Unlike single-turn input-output models, the proposed framework is designed to allow teachers to intervene at three distinct stages, shifting the interaction from post hoc adjustment toward more progressive involvement.

Moreover, the observed moderation by model scale and reasoning capability reveals that the framework's effectiveness is not uniform across evaluation dimensions. Smaller models can achieve structural completeness comparable to larger models when guided by theory-grounded prompts, making them viable for contexts where basic instructional structure is the primary need. However, deeper pedagogical qualities, such as cognitive level structuring and interdisciplinary integration, scale with reasoning capability, suggesting that different educational use cases may optimally pair with different model capacities.

In addition, the dual-track evaluation design contributes a methodological reference for LLM-as-Judge research in educational contexts. The strong rank-order agreement between human and LLM evaluations supports the viability of automated assessment for relative comparisons, while the systematic upward bias (+0.43) underscores the necessity of human calibration when absolute quality thresholds are at stake.

7.3. Practical Implications

The findings of this study have several practical implications for teachers, instructional designers, and educational institutions seeking to integrate LLMs into lesson planning workflows, specifically for Computer Science courses where lesson planning is conducted in English using one LLM family. The proposed framework offers a structurally complete prompting strategy that requires no model fine-tuning or technical expertise. Teachers can adopt the three-step structure and adapt the theory-based prompts to diverse teaching standards, student needs, classroom constraints, assessment requirements, and local culture.

Beyond serving immediate instructional needs, the process of editing and refining prompts can itself build teachers' AI literacy and teaching competency [61,62].

Moreover, the comparison between ACT-R and Gagné provides practical guidance for prompt selection. For educators seeking a detailed, structured sequence, Gagné's five-stage design performs well out of the box. For those preferring a simpler three-stage division, adding problem-chain guidance can substantially improve cognitive level structuring and interdisciplinary integration. This flexibility allows users to choose the most appropriate prompt variant based on their familiarity with pedagogical theories and the specific learning objectives of the lesson.

The moderation effect of model scale also carries practical relevance for resource-constrained institutions. Smaller models, such as the 32B distilled version used in this study, can produce structurally complete lesson plans when paired with the framework. Schools or universities without access to large-scale commercial LLMs can deploy smaller, locally hosted models combined with the proposed prompts to obtain usable lesson plans, lowering the barrier to AI-assisted lesson planning in low-resource settings. However, institutions aiming for deeper pedagogical quality, particularly in cognitive structuring and interdisciplinary design, would benefit from pairing the framework with stronger reasoning models.

7.4. Limitations and Future Research Directions

Several limitations of this study should be noted. The evaluation was conducted using only one subject domain (Computer Science), with lesson plans generated in English. Although the pedagogical theories are domain-general, the generalizability of the framework to other subjects, educational levels, and languages remains to be tested. Future research should therefore replicate the experiments across diverse disciplines and instructional contexts to establish broader applicability.

Another limitation concerns the assessment of lesson plan quality. While this study incorporated human validation to complement the automated evaluation, the human subsample was limited to 30 plans rated by five teachers within a single discipline. Per-indicator human data lack sufficient statistical power for fine-grained decomposition of strategy effects, and the human raters were drawn from the same subject domain as the session topics. Although the human ratings provided convergent evidence supporting the LLM judge's relative rankings, the LLM-as-Judge results should be considered preliminary until replicated with larger and more diverse human evaluator pools. Larger-scale human evaluations with raters from multiple disciplines, and classroom trials that test whether improved plan quality translates into better student learning outcomes, are important next steps.

The study also focused on a single model family: DeepSeek. Although three variants were used to explore scale and reasoning effects, results may differ for other LLMs [63] such as ChatGPT, Claude, or LLaMA-based models. Cross-family replication studies are needed to confirm the generalizability of the findings.

In addition, the claim that the framework supports progressive co-design represents a design affordance rather than an empirically observed behavioral outcome. This study did not directly observe teachers using the system across multiple design stages. Future work should investigate actual teacher-system interaction patterns to validate whether the intended multi-stage intervention translates into genuine collaborative design behavior.

Finally, the current framework assumes a single teacher-model interaction. More complex scenarios involving iterative co-design sessions, multiple teachers collaborating, or integration with learning management systems remain unexplored. Hu et al. [64] proposed a "Lesson Inquiry" model in which AI facilitates collaborative teaching research among

teachers, demonstrating that human–AI synergy can extend beyond individual lesson design to collective professional development. Future work could develop interactive interfaces that make the three-step intervention more intuitive, incorporate student feedback or learning analytics as a closed-loop optimization signal, and investigate the combination of the proposed prompt framework with fine-tuning or retrieval-augmented generation for further quality gains. Longitudinal studies in real classroom settings would also help validate the long-term impact of the framework on teaching practice and student achievement.

Author Contributions: Conceptualization, Y.L., Y.C. and W.L.; methodology, Y.L., Y.C. and W.L.; software, Y.L. and W.L.; validation, Y.L. and W.L.; formal analysis, Y.L., Y.C. and W.L.; investigation, Y.L. and W.L.; resources, Y.L., Y.C. and W.L.; data curation, Y.L., Y.C. and W.L.; writing—original draft preparation, Y.L.; writing—review and editing, Y.L., Y.C. and W.L.; visualization, Y.L.; supervision, Y.C.; project administration, Y.L. and Y.C.; funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Social Science Fund of China (Grant No. BJA240166), administered by the National Office for Philosophy and Social Sciences.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
LLM	Large Language Model
AIED	Artificial Intelligence in Education
COT	Chain-of-Thought
PCK	Pedagogical Content Knowledge
ACT-R	Adaptive Control of Thought-Rational
TCP	Transmission Control Protocol
JSON	JavaScript Object Notation
API	Application Programming Interface

Appendix A. Relevant Prompts

Appendix A.1. Baseline Prompt

Task Description: You are now a senior university teacher specializing in computer science. Please design a lesson plan for the topic {lesson_name} in the course {course_name}.

Appendix A.2. ACTR-Base Prompt

Step 1. Teaching Objectives Design:

You are a senior university teacher specializing in computer science and are proficient in Bloom’s Taxonomy. Based on the given course name and session topic, analyze the related teaching content in detail and generate structured instructional objectives in the following JSON format:

Element Specification: Each objective must include:

Bloom’s cognitive level verb

Specific knowledge point

Format Requirement: Use a numbered list. Each objective corresponds to a learning task for one knowledge point. There is no limit to the number. Example:

```
{"Teaching_Target": [
```

```

    "1. [level verb] + [knowledge point]",
    "2. [level verb] + [knowledge point]",
    "3. [level verb] + [knowledge point]",
    ...]
  }
  Course name: {course_name}
  Session topic: {session_topic}
  3.Example output:
  {"Teaching_Target": [
    "1. Master the working principle of TCP.",
    "2. Memorize the four-way handshake process of TCP.",
    "3. Deeply understand TCP congestion control.",
    ...]
  }

```

Output the structured result directly, without any additional information.

Step 2. Teaching Unit Design:

Task Background: You are now a senior university teacher specializing in computer science, designing a lesson plan for the {session_topic} in the {course_name}.

Reference Theory: The ACT-R theory describes the three stages students go through in the cognitive process of acquiring complex knowledge.

(1) **Declarative Stage:** Students primarily receive new knowledge to form a preliminary understanding. In this stage, students acquire declarative knowledge (e.g., facts, concepts) through teacher explanations or textbooks. Declarative knowledge is stored as chunks. Students need to construct knowledge structures through contextual anchoring and example comparison.

(2) **Procedural Stage:** Students begin to transform declarative knowledge into procedural knowledge, forming production rules through practice. This stage involves variant training and error correction. Students reduce errors and increase speed by continuously applying knowledge. The key is knowledge compilation, integrating fragmented declarative knowledge into executable rules (e.g., IF-THEN production rules). Cognitive load gradually decreases, but frequent feedback is needed to adjust rules.

(3) **Automation Stage:** Knowledge becomes fully proceduralized. Students can quickly and automatically retrieve and apply knowledge. Through extensive practice and cross-domain transfer, students reach an automated level. This stage involves myelination, strengthening neural pathways and speeding up responses. Cognitive load is minimal, allowing efficient handling of complex tasks.

Task Description: Based on the given instructional tasks and topic, design detailed teaching unit topics for the three stages.

Output Requirements:

Each stage contains multiple teaching units. The unit topics should align with the purpose and requirements of their respective stage.

Ensure a progressive knowledge sequence between units, avoiding jumps, and conforming to cognitive logic.

Output the structured result directly, without any other markers or explanations.

Output Format Reference:

```

{
  phase1_name: [Unit 1: Topic, Unit 2: Topic, ...]
  phase2_name: [Unit 1: Topic, Unit 2: Topic, ...]
  phase3_name: [Unit 1: Topic, Unit 2: Topic, ...]
}

```

Output the structured result directly, without any additional information.

Step 3. Teaching Activity Design (Using the Automation Stage of ACT-R as an example):

Task Background: You are now a senior university teacher specializing in computer science, designing a lesson plan for the {lesson_name} in the {course_name}. A complete three-stage unit plan is already available.

Reference Theory: (Same ACT-R theory description as in Step 2).

Task Description: The input will contain unit plans for multiple stages. Based on the given teaching unit arrangement, design and output teaching activities for each unit in the Automation Stage. Your goal in this stage is to facilitate the automated transfer of knowledge by designing complex, integrated scenario problems, cross-domain mapping, or timed challenges for complex tasks. This promotes the internalization and flexible application of previously learned knowledge. Output the detailed teaching plan for this stage.

Output Requirements:

Design a detailed teaching process for each unit in the Automation Stage.

Each teaching activity needs to elaborate on the actual teaching process.

Output the structured result directly, without any other markers or explanations.

Format requirements:

```
{
  phase3_name: {
    "Unit 1. Topic": [
      1. activity_name: Teaching Process Description,
      2. activity_name: Teaching Process Description,
      ...
    ],
    "Unit 2. Topic": [
      1. activity_name: Teaching Process Description,
      2. activity_name: Teaching Process Description,
      ...
    ]
  },
}
```

Output the structured result directly, without any additional information.

Appendix A.3. Gagné-Base Prompt

Step 1. Teaching Objectives Design: Same as ACTR-Base Step 1.

Step 2. Teaching Unit Design:

You are now a senior university teacher specializing in computer science, designing a lesson plan for the {session_topic} in the {course_name}. Based on the given instructional objectives, design teaching units according to the following five stages:

Pre-class Introduction Stage (Stimulate interest + Review prior knowledge)

New Knowledge Presentation Stage (Explain core concepts)

New Knowledge Consolidation & Connection to Prior Knowledge Stage (Practice + Association)

Application and Transfer Stage (Practice + Higher-order thinking)

Summary and Review Stage (Reinforce memory + Check for gaps)

Requirements:

Each stage contains multiple teaching units.

Output the topic of each unit.

Ensure a progressive knowledge sequence between unit topics, avoiding jumps, and conforming to cognitive logic.

Output Format Reference:

```
{
    phase1_name: [Unit 1: Topic, Unit 2: Topic, ...]
    phase2_name: [Unit 1: Topic, Unit 2: Topic, ...]
    ...
}
```

Output Example:

```
{
    Pre-class Introduction Stage: [
        Unit 1: Introduce the lesson, stimulate interest,
        ...
    ],
    New Knowledge Explanation Stage: [
        Unit 1: Explain the layered network architecture,
        Unit 2: Elaborate on the basic characteristics of the TCP,
        Unit 3: Briefly introduce the TCP connection establishment and termination process,
        ...
    ],
    ...
}
```

Output the structured result directly, without any additional information.

Step 3. Teaching Activity Design (Using the Pre-class Introduction Stage of Gagné's Five Stages as an example):

Task Background: You are now a senior university teacher specializing in computer science, designing a lesson plan for the {session_topic} in the {course_name}.

Known Stages: The lesson is divided into the following five stages: 1. Pre-class Introduction, 2. New Knowledge Explanation, 3. New Knowledge Consolidation & Connection to Prior Knowledge, 4. Application and Enhancement, 5. Summary and Review.

Current Stage: You are currently designing teaching activities for the Pre-class Introduction Stage. The teaching task of this stage is to introduce the knowledge point, activate students' prior knowledge, quickly focus their attention, and establish connections between old and new knowledge, stimulating interest while naturally transitioning to new knowledge.

Output Requirements:

Design a detailed teaching process for each unit in the Pre-class Introduction Stage.

Each teaching activity needs to elaborate on the actual teaching process.

Output the structured result directly, without any other markers or explanations.

Format requirements:

```
{
    phase1_name: {
        "Unit 1. Topic": [
            1. activity_name: Teaching Process Description,
            2. activity_name: Teaching Process Description,
            ...
        ],
        "Unit 2. Topic": [
            1. activity_name: Teaching Process Description,
```

```

        2. activity_name: Teaching Process Description,
        ...
    ]
},
}

```

Output the structured result directly, without any additional information.

Appendix A.4. ACTR-Probchain Prompt

Step 1. Teaching Objectives Design: Same as ACTR-Base Step 1.

Step 2. Teaching Unit Design: Same as ACTR-Base Step 2.

Step 3. Teaching Activity Design (Using the Declarative Stage of ACT-R as an example):

Task Background: You are now a senior university teacher specializing in computer science, designing a lesson plan for the {session_topic} in the {course_name}. According to ACT-R theory, the lesson is divided into three stages: (1) Declarative Stage, (2) Procedural Stage, (3) Automation Stage. (The detailed theory description is the same as in Appendix A.2 Step 2 and is omitted here for brevity).

Current Stage: You are currently designing teaching activities for the Declarative Stage. In this stage, the teacher focuses on knowledge structuring, using tools like mind maps or visualizations to present the knowledge framework. Guide students to gradually understand the new knowledge from basic to challenging tasks. In this stage, the following three problem chains are recommended for design:

<| Recommended Problem Chains Start |>

Introductory Problem Chain: Start from life scenarios, introduce the topic through a series of consecutive questions to arouse student attention. Questions can be created from students' prior knowledge and life scenarios.

Transfer Problem Chain: Promote knowledge formation and consolidation through multiple questions involving transfer to similar scenarios or disciplines, enhancing the ability to generalize problem-solving.

Discrepancy Problem Chain: Use multiple questions that contradict students' preconceptions or attempts to create cognitive imbalance, thereby activating prior knowledge networks and reshaping scientific concepts. Deepen memory through cognitive conflict caused by unexpected results.

<| Recommended Problem Chains End |>

Task Description: The input will contain unit plans for multiple stages. Based on the given teaching unit arrangement, and with reference to the recommended problem chains and the teaching tasks of the Declarative Stage, match an appropriate problem chain type for each teaching unit in the Declarative Stage, design the problem chain, and design the teaching process around it.

Requirements:

Each problem chain must contain two or more progressive or consecutive specific questions.

Output format example:

```

{
    phase1_name: {
        "Unit 1. Topic": [
            Problem Chain: Question1? Question2? ... ?,
            1. activity_name: Teaching Process Description,
            2. activity_name: Teaching Process Description,
        ],
    }
}

```

```

        "Unit 2. Topic": [
        Problem Chain: Question1? Question2? ... ?,
        1. activity_name: Teaching Process Description,
        2. activity_name: Teaching Process Description,
        ],
        ...
    },
}

```

Output the structured result directly, without any additional information.

Appendix A.5. Gagné-Probchain Prompt

Step 1. Teaching Objectives Design: Same as ACTR-Base Step 1.

Step 2. Teaching Unit Design: Same as Gagné-Base Step 2.

Step 3. Teaching Activity Design (Using the Pre-class Introduction Stage of Gagné's Five Stages as an example):

Task Background: You are now a senior university teacher specializing in computer science, designing a lesson plan for the {session_topic} in the {course_name}.

Known Stages: The lesson is divided into the following five stages: 1. Pre-class Introduction, 2. New Knowledge Presentation, 3. New Knowledge Consolidation & Connection to Prior Knowledge, 4. Application and Transfer, 5. Summary and Review.

Current Stage: You are currently designing teaching activities for the Pre-class Introduction Stage. The teaching task of this stage is to introduce the knowledge point, activate students' prior knowledge, quickly focus their attention, and establish connections between old and new knowledge, stimulating interest while naturally transitioning to new knowledge. In this stage, please primarily refer to these three types of problem chains for design:

<| Reference Problem Chains Start |>

Introductory Problem Chain: Introduce the topic through a series of consecutive questions to arouse student attention. Questions can be created from students' prior knowledge and life scenarios to introduce the new knowledge point.

Transfer Problem Chain: Promote knowledge formation and consolidation through multiple questions involving transfer to similar scenarios or disciplines, enhancing the ability to apply knowledge to solve problems in different contexts.

Discrepancy Problem Chain: A series of questions that contradict students' preconceptions or attempts, creating cognitive conflict through unexpected results to deepen memory.

<| Reference Problem Chains End |>

Task Description: The input will contain unit plans for multiple stages. Based on the given teaching unit arrangement, and with reference to the given problem chains and the teaching tasks of the Pre-class Introduction Stage, select an appropriate problem chain type for each teaching unit in the Pre-class Introduction Stage, design the problem chain, and design the teaching process around it.

Requirements:

Each problem chain must contain two or more progressive or consecutive specific questions.

Organize the output in the following format:

```

{
    phase1_name: {
        "Unit 1. Topic": [
        Problem Chain: Question1? Question2? ... ?,

```

```

        1. activity_name: Teaching Process Description,
        2. activity_name: Teaching Process Description,
    ],
    "Unit 2. Topic": [
        Problem Chain: Question1? Question2? ... ?,
        1. activity_name: Teaching Process Description,
        2. activity_name: Teaching Process Description,
    ],
    ...
},
}

```

Output the structured result directly, without any additional information.

Appendix B. Sample Outputs of Prompts

```

{
  "Teaching_Target": [
    "1. Understand the transport layer positioning of TCP in network architecture",
    "2. Understand the connection-oriented service characteristics of TCP",
    "3. Master the header format and field functions of TCP segments",
    "4. Analyze the process of establishing a connection with three-way handshake and state transitions",
    "5. Analyze the process of releasing a connection with four-way wavehand and state transitions",
    "6. Understand the mechanism of sequence numbers and acknowledgment numbers in reliable transmission",
    "7. Master the principle of flow control implemented by sliding window",
    "8. Learn congestion control algorithms (slow start / congestion avoidance / fast recovery)",
    "9. Distinguish the goal differences between flow control and congestion control",
    "10. Master the trigger conditions for timeout retransmission and fast retransmission",
    "11. Understand the working mechanisms of cumulative acknowledgment and selective acknowledgment",
    "12. Apply TCP state transition diagrams to analyze abnormal network scenarios",
    "13. Comprehensively apply TCP mechanisms to explain actual data transmission phenomena"
  ]
}

```

Figure A1. Sample output of teaching objectives.

```

{
  "Declarative Stage": [
    "Unit 1: Transport layer positioning and core responsibilities of the TCP protocol in the OSI/RM architecture",
    "Unit 2: The relationship between connection-oriented service characteristics and the Socket abstract interface",
    "Unit 3: Visual analysis of TCP segment header format and field classification",
    "Unit 4: Basic process of connection establishment/release and cognition of state terminology",
    "Unit 5: Correspondence between basic concepts of reliable transmission and network service quality requirements"
  ],
  "Procedural Stage": [
    "Unit 1: Temporal modeling of state transitions during the three-way handshake process",
    "Unit 2: Deduction of abnormal scenarios in state transitions during the four-way handshake",
    "Unit 3: Mapping relationship between sequence number space and byte stream order",
    "Unit 4: Visualized dynamic demonstration of the sliding window mechanism",
    "Unit 5: Derivation of phase transition conditions for congestion control algorithms",
    "Unit 6: Linkage between acknowledgment mechanisms and timeout retransmission timers"
  ],
  "Autonomous Stage": [
    "Unit 1: Analysis of the synergistic effect of flow control and congestion control",
    "Unit 2: Mathematical relationship modeling of fast retransmission and duplicate ACKs",
    "Unit 3: Bitmap encoding analysis of the Selective Acknowledgment (SACK) option",
    "Unit 4: Application of state transition diagrams in connection reset scenarios",
    "Unit 5: Bandwidth-delay product optimization of dynamic congestion window adjustment",
    "Unit 6: Reverse analysis of complete TCP sessions using Wireshark instances"
  ]
}

```

Figure A2. Sample output of instructional units.

```

{
  "Declarative_stage": [
    {
      "Unit 1": [
        {
          "title": "Transport layer positioning and core responsibilities of TCP in OSI/RM.",
          "problem_chain": [
            "Q1: Why do we need a transport layer when the network layer already provides host-to-host communication?",
            "Q2: If IP is inherently unreliable, how can upper-layer applications ensure data integrity?"
          ],
          "activities": [
            "Step 1: Teacher projects OSI/RM diagram, asks Q1. Students discuss in pairs (2 min).",
            "Step 2: Teacher plays animation of IP packet loss and out-of-order delivery, asks Q2. Students identify the gap between IP service and application needs.",
            "Step 3: Teacher introduces TCP core responsibilities (segmentation, reassembly, error control, flow control, congestion control). Students take notes.",
            "Step 4: Teacher maps each TCP responsibility to a specific IP shortcoming. [Total: 12 min]"
          ]
        }
      ]
    }
  ]
}

```

Figure A3. Sample output of problem-chain guided activity design.

References

- Chiu, T.K.F.; Xia, Q.; Zhou, X.; Chai, C.S.; Cheng, M. Systematic Literature Review on Opportunities, Challenges, and Future Research Recommendations of Artificial Intelligence in Education. *Comput. Educ. Artif. Intell.* **2023**, *4*, 100118. [\[CrossRef\]](#)
- Fan, H.; Chen, G.; Wang, X.; Peng, Z. LessonPlanner: Assisting Novice Teachers to Prepare Pedagogy-Driven Lesson Plans with Large Language Models. In *Proceedings of the ACM Conference on Learning @ Scale, Atlanta, GA, USA, 18–20 July 2024*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 1–20.
- Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback. *arXiv* **2022**, arXiv:2203.02155. [\[CrossRef\]](#)
- Davis, R.O.; Lee, Y.J. Prompt: ChatGPT, Create My Course, Please! *Educ. Sci.* **2023**, *14*, 24. [\[CrossRef\]](#)
- Van Den Berg, G.; Du Plessis, E. ChatGPT and Generative AI: Possibilities for Its Contribution to Lesson Planning, Critical Thinking and Openness in Teacher Education. *Educ. Sci.* **2023**, *13*, 998. [\[CrossRef\]](#)
- Hu, B.; Zheng, L.; Zhu, J.; Ding, L.; Wang, Y.; Gu, X. Teaching Plan Generation and Evaluation With GPT-4: Unleashing the Potential of LLM in Instructional Design. *IEEE Trans. Learn. Technol.* **2024**, *17*, 1445–1459. [\[CrossRef\]](#)
- Karaman, M.R.; Göksu, İ. Are Lesson Plans Created by ChatGPT More Effective? An Experimental Study. *Int. J. Technol. Educ.* **2024**, *7*, 107–127. [\[CrossRef\]](#)
- Zheng, Y.; Li, X.; Huang, Y.; Liang, Q.; Guo, T.; Hou, M.; Gao, B.; Tian, M.; Liu, Z.; Luo, W. Automatic Lesson Plan Generation via Large Language Models with Self-Critique Prompting. In *Proceedings of the Artificial Intelligence in Education, Recife, Brazil, 8–12 July 2024*; Springer Nature: Cham, Switzerland, 2024; pp. 163–178.
- Standen, P.J.; Brown, D.J.; Taheri, M.; Galvez Trigo, M.J.; Boulton, H.; Burton, A.; Hallewell, M.J.; Lathe, J.G.; Shopland, N.; Blanco Gonzalez, M.A.; et al. An Evaluation of an Adaptive Learning System Based on Multimodal Affect Recognition for Learners with Intellectual Disabilities. *Br. J. Educ. Tech.* **2020**, *51*, 1748–1765. [\[CrossRef\]](#)
- Martinez-Maldonado, R.; Clayphan, A.; Yacef, K.; Kay, J. MTFeedback: Providing Notifications to Enhance Teacher Awareness of Small Group Work in the Classroom. *IEEE Trans. Learn. Technol.* **2015**, *8*, 187–200. [\[CrossRef\]](#)
- Fu, S.; Gu, H.; Yang, B. The Affordances of AI-enabled Automatic Scoring Applications on Learners' Continuous Learning Intention: An Empirical Study in China. *Br. J. Educ. Tech.* **2020**, *51*, 1674–1692. [\[CrossRef\]](#)
- Zawacki-Richter, O.; Marín, V.I.; Bond, M.; Gouverneur, F. Systematic Review of Research on Artificial Intelligence Applications in Higher Education—Where Are the Educators? *Int. J. Educ. Technol. High. Educ.* **2019**, *16*, 39. [\[CrossRef\]](#)
- Forero-Corba, W.; Negre Bennasar, F. Técnicas y Aplicaciones Del Machine Learning e Inteligencia Artificial En Educación: Una Revisión Sistemática. *Rev. Iberoam. Educ. Distancia* **2023**, *27*, 209–253. [\[CrossRef\]](#)
- Brusilovsky, P. AI in Education, Learner Control, and Human-AI Collaboration. *Int. J. Artif. Intell. Educ.* **2024**, *34*, 122–135. [\[CrossRef\]](#)
- Naveed, H.; Khan, A.U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Akhtar, N.; Barnes, N.; Mian, A. A Comprehensive Overview of Large Language Models. *ACM Trans. Intell. Syst. Technol.* **2025**, *16*, 106. [\[CrossRef\]](#)
- Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *ACM Comput. Surv.* **2023**, *55*, 105. [\[CrossRef\]](#)
- Sahoo, P.; Singh, A.K.; Saha, S.; Jain, V.; Mondal, S.; Chadha, A. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. *arXiv* **2024**, arXiv:2402.07927.
- Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.H.; Le, Q.V.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022*; Curran Associates, Inc.: Red Hook, NY, USA, 2022; Volume 35, pp. 24824–24837.
- Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models Are Few-Shot Learners. In *Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; pp. 1877–1901.

20. ElSayary, A.; Kuhail, M.A.; Hojeij, Z. Examining the Role of Prompt Engineering in Utilizing Generative AI Tools for Lesson Planning: Insights From Teachers' Experiences and Perceptions. *Hum. Behav. Emerg. Technol.* **2025**, *2025*, 9986139. [\[CrossRef\]](#)
21. Peikos, G.; Stavrou, D. ChatGPT for Science Lesson Planning: An Exploratory Study Based on Pedagogical Content Knowledge. *Educ. Sci.* **2025**, *15*, 338. [\[CrossRef\]](#)
22. Bai, S.; Lo, C.K.; Yang, C. Enhancing Instructional Design Learning: A Comparative Study of Scaffolding by a 5E Instructional Model-Informed Artificial Intelligence Chatbot and a Human Teacher. *Interact. Learn. Environ.* **2025**, *33*, 2738–2757. [\[CrossRef\]](#)
23. Hu, B.; Zhu, J.; Pei, Y.; Gu, X. Exploring the Potential of LLM to Enhance Teaching Plans through Teaching Simulation. *npj Sci. Learn.* **2025**, *10*, 7. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Cai, H.; Han, B.; Sun, J.; Li, X.; Wong, L.-H. Harnessing AI for Teacher Education to Promote Inclusive Education: Investigating the Effects of ChatGPT-Supported Lesson Plan Critiques on the Development of Pre-Service Teachers' Lesson Planning Skills. *Internet High. Educ.* **2025**, *67*, 101022. [\[CrossRef\]](#)
25. Tian, F. Building an Instructional Design Model for Human-AI Collaboration Grounded in Activity Theory. In *Proceedings of the 15th International Conference on Education Technology and Computers, Barcelona, Spain, 26 September 2023*; ACM: New York, NY, USA, 2023; pp. 33–39.
26. Luna Chontal, G.; Melendez-Armenta, R.A.; Degante-Aguilar, E.; Fernández-Domínguez, F.J. Task Automation and Instructional Planning Support with Large Language Models: A Systematic Review. *Front. Educ.* **2026**, *11*, 1733861. [\[CrossRef\]](#)
27. Qian, Y. Prompt Engineering in Education: A Systematic Review of Approaches and Educational Applications. *J. Educ. Comput. Res.* **2025**, *63*, 1782–1818. [\[CrossRef\]](#)
28. Francom, G. Ten Steps to Complex Learning: A Systematic Approach to Four-Component Instructional Design (3rd Ed.), by Jeroen J. G. van Merriënboer and Paul A. Kirschner. *TechTrends* **2018**, *62*, 204–205. [\[CrossRef\]](#)
29. Padovano, A.; Cardamone, M. Towards Human-AI Collaboration in the Competency-Based Curriculum Development Process: The Case of Industrial Engineering and Management Education. *Comput. Educ. Artif. Intell.* **2024**, *7*, 100256. [\[CrossRef\]](#)
30. Tseng, Y.-C.; Lin, Y.-H. Enhancing English as a Foreign Language (EFL) Learners' Writing with ChatGPT: A University-Level Course Design. *Electron. J. E-Learn.* **2024**, *22*, 78–97. [\[CrossRef\]](#)
31. Frerejean, J.; Van Geel, M.; Keuning, T.; Dolmans, D.; Van Merriënboer, J.J.G.; Visscher, A.J. Ten Steps to 4C/ID: Training Differentiation Skills in a Professional Development Program for Teachers. *Instr. Sci.* **2021**, *49*, 395–418. [\[CrossRef\]](#)
32. Gagné, R.M. *The Conditions of Learning*, 4th ed.; Holt, Rinehart and Winston: New York, NY, USA, 1985.
33. Biggs, J. Enhancing Teaching through Constructive Alignment. *High. Educ.* **1996**, *32*, 347–364. [\[CrossRef\]](#)
34. Squalli Houssaini, M.; Aboutajeddine, A.; Toughrai, I.; Ibrahim, A. Development of a Design Course for Medical Curriculum: Using Design Thinking as an Instructional Design Method Empowered by Constructive Alignment and Generative AI. *Think. Ski. Creat.* **2024**, *52*, 101491. [\[CrossRef\]](#)
35. Dogan, S. A Human-AI Hybrid Co-Design Model: Planning Effective Instruction with ChatGPT. *Asian J. Distance Educ.* **2025**, *20*, 94–119. [\[CrossRef\]](#)
36. Mollick, E.; Mollick, L. Assigning AI: Seven Approaches for Students, with Prompts. *arXiv* **2023**, arXiv:2306.10052.
37. Wang, Z.; Mingzhuo, L.; Islam, A.Y.M.A. Reimagining Teacher-AI Co-Design in Learning Task Design: Trends and Perspectives. *Humanit. Soc. Sci. Commun.* **2026**, *13*, 757. [\[CrossRef\]](#)
38. Bloom, B.S.; Engelhart, M.D.; Furst, E.J.; Hill, W.H.; Krathwohl, D.R. Taxonomy of Educational Objectives: The Classification of Educational Goals. In *Handbook I: Cognitive Domain*; David McKay: New York, NY, USA, 1956.
39. Anderson, L.W.; Krathwohl, D.R.; Airasian, P.W.; Cruikshank, K.A.; Mayer, R.E.; Pintrich, P.R.; Raths, J.; Wittrock, M.C. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*; Longman: New York, NY, USA, 2001.
40. Reigeluth, C.M. *Instructional-Design Theories and Models: A New Paradigm of Instructional Theory, Volume II.*; Taylor and Francis: Hoboken, NJ, USA, 2012.
41. Anderson, J.R. *The Architecture of Cognition*; Harvard University Press: Cambridge, MA, USA, 1983.
42. Anderson, J.R.; Bothell, D.; Byrne, M.D.; Douglass, S.; Lebiere, C.; Qin, Y. An Integrated Theory of the Mind. *Psychol. Rev.* **2004**, *111*, 1036–1060. [\[CrossRef\]](#) [\[PubMed\]](#)
43. Anderson, J.R.; Corbett, A.T.; Koedinger, K.R.; Pelletier, R. Cognitive Tutors: Lessons Learned. *J. Learn. Sci.* **1995**, *4*, 167–207. [\[CrossRef\]](#) [\[PubMed\]](#)
44. Gagné, R.M. (Ed.) *Principles of Instructional Design*, 5th ed.; Wadsworth: Belmont, CA, USA, 2011.
45. Wang, H.X. Types and teaching functions of problem chain: Taking chemistry teaching as an example. *Educ. Sci. Res.* **2010**, *5*, 50–54.
46. Chen, Y.F. Constructing big unit teaching design based on problem chain: Taking transportation and regional development as an example. *Geogr. Teach.* **2025**, *3*, 24–28.
47. DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Wang, P.; Zhu, Q.; Xu, R.; Zhang, R.; Ma, S.; et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv* **2025**, arXiv:2501.12948.

48. Holtzman, A.; Buys, J.; Du, L.; Forbes, M.; Choi, Y. The Curious Case of Neural Text Degeneration. *arXiv* **2019**, arXiv:1904.09751.
49. Clark, H.-B.; Dowland, M.; Benton, L.; Budai, R.; Keskin, I.K.; Searle, E.; Gregory, M.; Hodierne, M.; Roberts, J. Auto-Evaluation: A Critical Measure in Driving Improvements in Quality and Safety of AI-Generated Lesson Resources. *arXiv* **2025**, arXiv:2502.10410.
50. Feng, G.; Zhang, B.; Gu, Y.; Ye, H.; He, D.; Wang, L. Towards Revealing the Mystery behind Chain of Thought: A Theoretical Perspective. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 70757–70798. [[CrossRef](#)]
51. Koo, T.K.; Li, M.Y. A Guideline of Selecting and Reporting Intraclass Correlation Coefficients for Reliability Research. *J. Chiropr. Med.* **2016**, *15*, 155–163. [[CrossRef](#)] [[PubMed](#)]
52. Vallat, R. Pingouin: Statistics in Python. *J. Open Source Softw.* **2018**, *3*, 1026. [[CrossRef](#)]
53. Jung, Y.; Wise, A.F. How Students Engage with Learning Analytics: Access, Action-Taking, and Learning Routines with Message-Based Information to Support Collaborative Annotation. *Comput. Educ.* **2025**, *232*, 105280. [[CrossRef](#)]
54. Zhang, X.; Zhang, C.; Sun, J.; Xiao, J.; Yang, Y.; Luo, Y. EduPlanner: LLM-Based Multiagent Systems for Customized and Intelligent Instructional Design. *IEEE Trans. Learn. Technol.* **2025**, *18*, 416–427. [[CrossRef](#)]
55. Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.P.; et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. *arXiv* **2023**, arXiv:2306.05685.
56. Sweller, J.; Van Merriënboer, J.J.G.; Paas, F. Cognitive Architecture and Instructional Design: 20 Years Later. *Educ. Psychol. Rev.* **2019**, *31*, 261–292. [[CrossRef](#)]
57. Reiser, B.J. Scaffolding Complex Learning: The Mechanisms of Structuring and Problematising Student Work. *J. Learn. Sci.* **2004**, *13*, 273–304. [[CrossRef](#)] [[PubMed](#)]
58. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; et al. Emergent Abilities of Large Language Models. *arXiv* **2022**, arXiv:2206.07682.
59. Thakur, A.S.; Choudhary, K.; Ramayapally, V.S.; Vaidyanathan, S.; Hupkes, D. Judging the Judges: Evaluating Alignment and Vulnerabilities in LLMs-as-Judges. *arXiv* **2024**, arXiv:2406.12624.
60. Dourvas, I.; Kokkonis, G.; Kontogiannis, S. Reconceptualizing Prompt Engineering as Reflective Professional Practice: A Framework for Teacher Development. *Electronics* **2026**, *15*, 930. [[CrossRef](#)]
61. Zhou, J.; Si, L.; Liu, X. ‘A Good Question Is Half the Answer:’ A Learning-Oriented K-12 Prompt Literacy Competency Cultivation Framework. *Inf. Res. Int. Electron. J.* **2026**, *31*, 1597–1607. [[CrossRef](#)]
62. Wang, X.; Niu, J.; Fang, B.; Han, G.; He, J. Empowering Teachers’ Professional Development with LLMs: An Empirical Study of Developing Teachers’ Competency for Instructional Design in Blended Learning. *Teach. Teach. Educ.* **2025**, *165*, 105091. [[CrossRef](#)]
63. Xue, M.; Xiao, X.; Liu, Y.; Wilson, M. On the Consistency of Automatic Scoring with Large Language Models. *Educ. Psychol. Meas.* **2026**, online first. [[CrossRef](#)] [[PubMed](#)]
64. Hu, X.Y.; Xie, Y.Q.; Lu, J.R.; Lin, Z.R. Questioning: Construction and application of AIED human-machine collaborative teaching research model for promoting wisdom generation. *Mod. Distance Educ.* **2025**, *23*, 23–32. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.