From Knowing to Teaching: Scaffolding Pedagogical Decisions for LLM Agent

Yucheng Wang[1], Shen Yang[1], Jifan Yu[2], Haoxuan Li[3], Joy Jia Yin Lim[1], Daniel Zhang-Li[1], Huiqin Liu[2], Lei Hou[1], Juanzi Li[1], Bin Xu[1]*

[1]Department of Computer Science and Technology, Tsinghua University
[2]Institute of Education, Tsinghua University [3]College of AI, Tsinghua University
Correspondence: xubin@tsinghua.edu.cn

Abstract

Knowing and teaching differ fundamentally: effective instruction requires transforming knowledge into forms learners can grasp. Large language models, when asked to generate lessons (a concrete form of teaching), produce content lacking pedagogical depth. We trace this failure to three decisions that expert teachers make: *selecting* content by recognizing each source's instructional role, *sequencing* topics so foundations precede applications, and *synthesizing* components into a unified whole. To scaffold these decisions, we introduce **TeachCraft**, a framework with three agents: Explorer classifies sources by pedagogical intent to guide selection; Planner orders objectives from foundational to advanced; Generator produces lesson materials through a schema that ensures consistency across components. To evaluate this approach, we construct LESSONBENCH, 40 expert-designed lessons paired with two to five heterogeneous source documents, on which TeachCraft achieves 67.8% win rate in human evaluation and 79.6% in LLM-based evaluation against eight baselines, with ablations confirming that each decision contributes independently to overall lesson quality.[1]

1 Introduction

Large language models encode vast knowledge (Hendrycks et al., 2020), making them natural candidates for educational applications. Yet teaching requires more than knowledge retrieval; it demands what Shulman (1986) termed *Pedagogical Content Knowledge* (PCK), the expertise that enables teachers to transform what they know into forms learners can grasp. When asked to generate lessons, a concrete and measurable form of teaching, LLMs often produce content that looks polished but lacks pedagogical value, revealing a knowing-teaching gap, where gaining PCK cultivates human teachers.

Figure 1 illustrates this clear gap in a toy case of the sorting algorithm lesson. An expert teacher, handed a textbook and a style template, recognizes what each offers: core content and visual format. She makes a *selection* according to a pedagogical intent. An LLM, lacking such *intent awareness*, treats all inputs as interchangeable and may pull content from a template simply because it mentions sorting. The teacher arranges material of dependent topics matching prerequisites. This *sequencing* respects *cognitive progression* (Bloom et al., 1956), while the model skips the process. Finally, she ensures that every quiz question tests what the slides introduced, a *synthesis* demanding *cross-component coherence* (Mayer, 2009) that breaks down when an LLM asks about HeapSort, which the slides never introduced.

To bridge this knowing-teaching gap, prior work has explored various approaches: single-document systems that personalize content assuming well-structured inputs (Team et al., 2025), topic-driven generators that create lessons ungrounded in source materials (Yao et al., 2025), format converters that ignore pedagogical roles (Zheng et al., 2025a), and frameworks like ADDIE (Branch, 2009) that prescribe design phases without specifying the judgments within them. However, none model the three decisions distinguishing expert teaching: *selection* recognizing pedagogical intent, *sequencing* building cognitive progression, and *synthesis* ensuring cross-component coherence. While these decisions manifest across all teaching modalities—in Socratic dialogue, selection identifies misconceptions; sequencing orders questions by cognitive demand; synthesis ensures follow-ups build on student responses—structured lesson generation provides a concrete, evaluable setting in which all three produce measurable artifacts.

Therefore, we introduce **TeachCraft** to operationalize these decisions, formulating lesson creation as *multi-document lesson synthesis*: given

* Corresponding author.
[1]Source code is available at https://github.com/wyuc/TeachCraft.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 28778–28801
July 2-7, 2026

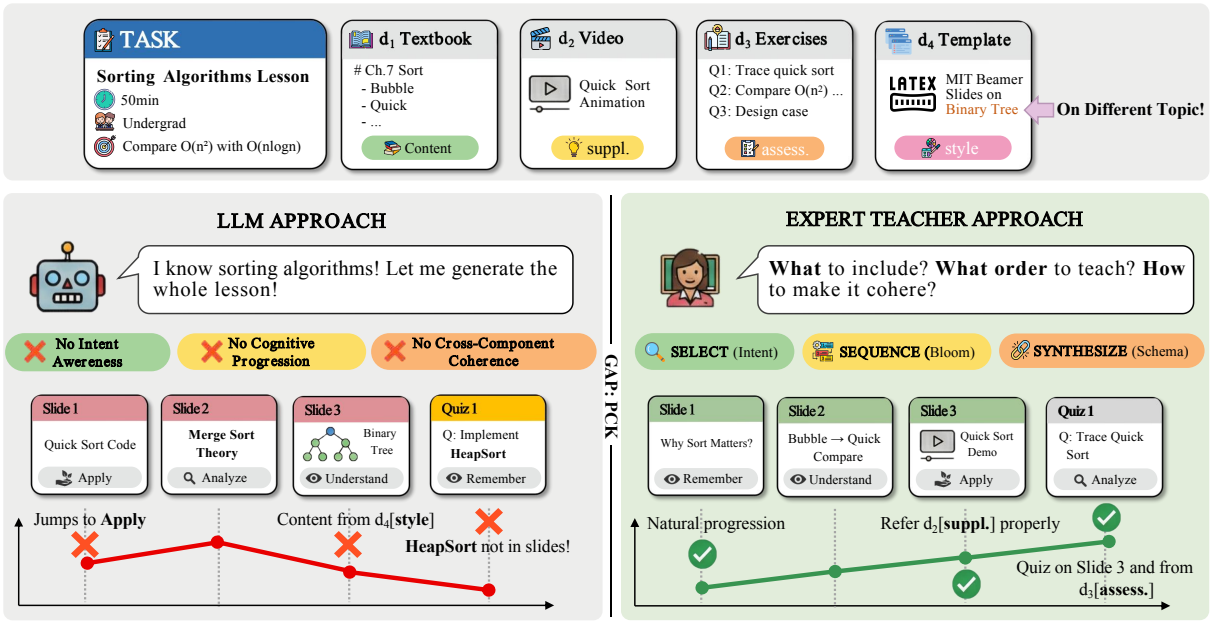

Figure 1: **The Knowing-Teaching Gap.** Given a sorting algorithms lesson task with intent-annotated source documents (Content, Supplementary, Assessment, Style), LLMs produce lessons with three characteristic failures: using style templates as content (no intent awareness), jumping to Apply-level material without foundations (no cognitive progression), and testing HeapSort never introduced in slides (no cross-component coherence). Expert teachers make three deliberate decisions that produce coherent lessons: *Selection*, *Sequencing*, and *Synthesis*. TeachCraft operationalizes these decisions through three specialized agents (Explorer, Planner, Generator).

heterogeneous documents $\mathcal{D}$ and requirements $\mathcal{R}$, produce a lesson $\mathcal{L}$ of slides, scripts, and quizzes that together span the core modalities of instruction. Three specialized agents mirror the teacher's decisions: **Explorer** classifies sources by pedagogical intent to guide selection; **Planner** sequences objectives from foundational to advanced following Bloom's taxonomy; **Generator** produces components through a unified schema ensuring cross-component consistency.

Evaluating lesson generation requires assessing pedagogical soundness and cross-component coherence, which existing benchmarks do not cover. We therefore construct LESSONBENCH, a pairwise comparison benchmark with human and LLM judges. TeachCraft achieves over **67%** win rate against eight baselines; ablations confirm each decision contributes independently. We further validate on SLIDESBENCH (Ge et al., 2025), an established slide generation benchmark, demonstrating competitive visual quality with full execution reliability.

Contributions.

- We propose a practical framework operationalizing Pedagogical Content Knowledge through three decisions (selection, sequencing, synthesis), introducing TeachCraft with specialized agents (Explorer, Planner, Generator) for each.
- We construct LESSONBENCH, a benchmark of 40 expert-designed lessons across four subject areas for multi-document lesson synthesis.
- TeachCraft achieves 67% human and 79% LLM judge win rates against eight baselines, with competitive visual quality on SLIDESBENCH; case study analysis reveals gradual cognitive progression more aligned with expert instruction.

2 Related Work

Our task builds on multi-document summarization (MDS), which combines information from multiple sources into coherent outputs (Nenkova et al., 2011). Early work recognized that effective MDS requires *information fusion*, integrating content across documents rather than merely selecting sentences (Barzilay et al., 1999). DeYoung et al. (2024) formalize this distinction: most systems perform *aggregation* (selecting and concatenating passages) rather than *synthesis* (producing genuinely new text). Educational content demands synthesis: a lesson on sorting algorithms should unify explanations from a textbook with examples from lecture notes, not merely juxtapose excerpts.

Large language models have advanced educa-

System	Multi-Doc	Ped. Ground.	Intent	Output
Learn Your Way	✗	✗	✗	Text
Instr. Agents	✗	ADDIE	✗	S
NotebookLM	✓	✗	✗	Audio
PPTAgent	✗	✗	✗	S
TeachCraft	✓	PCK+Bloom	✓	S+Sc+Q

Table 1: Comparison with prior systems. Output: S=Slides, Sc=Scripts, Q=Quizzes.

tional content generation along separate tracks. For slides, systems convert documents into presentations through edit-based refinement (Zheng et al., 2025a), code generation (Ge et al., 2025), or synchronized speech (Aggarwal and Bhand, 2025); earlier work like DocPres (Bandyopadhyay et al., 2024) established document-to-slide pipelines. For assessments, question generators target specific cognitive levels (Scaria et al., 2024) or apply controllable constraints (Li and Zhang, 2024). For lesson planning, Instructional Agents (Yao et al., 2025) coordinates multiple agents following the ADDIE framework; LessonPlanLM (Zheng et al., 2025b) augments generation with knowledge bases; Learn Your Way (Team et al., 2025) personalizes learning within a single textbook. Each makes progress, but they remain separate: slide generators do not consider assessment; question generators ignore what has been taught; planners work from topic names rather than source documents.

Recent products like NotebookLM (Google, 2024) attempt end-to-end synthesis from uploaded materials, yet lack explicit structure for pedagogical decisions. Table 1 summarizes how TeachCraft differs: it jointly processes heterogeneous sources, grounds generation in pedagogical theory, and produces integrated outputs (slides, scripts, quizzes).

3 TeachCraft

We formalize how TeachCraft bridges the knowing-teaching gap through three core pedagogical decisions (selection, sequencing, synthesis) implemented by specialized agents that mirror the expertise of experienced teachers (Figure 2).

3.1 Formative Insights from Educators

To ground our framework in pedagogical practice rather than engineering intuition, we conducted semi-structured interviews with three K-12 educators spanning Chinese Literature, English, and Biology, with 1–18 years of experience (details in Appendix A). Three consistent patterns emerged that motivated our design decisions:

Selection. Teachers described viewing materials through a *pedagogical lens*: "*After gathering materials, my next task is filtering. I filter based on curriculum standards and student readiness*" (T1). This intent-driven filtering—where each source serves a distinct instructional role—motivated our six-category intent taxonomy.

Sequencing. Teachers emphasized deliberate cognitive progression: "*Question chains guide students to think progressively deeper—from surface features to emotional changes, then to analyzing contradictions*" (T1). This aligns with Bloom's taxonomy and motivated Planner's objective ordering.

Synthesis. Teachers stressed cross-component coherence: "*My questions should follow one main thread. I must truly integrate these materials so that students experience a coherent lesson*" (T1). This motivated our schema-based generation ensuring consistency across slides, scripts, and quizzes.

3.2 Formulation

Given heterogeneous documents $\mathcal{D} = \{d_1, \ldots, d_n\}$ with pedagogical intents and natural language requirements $\mathcal{R}$, the goal is to produce a lesson $\mathcal{L} = (c_1, \ldots, c_m)$ where each component c_j is either a `Slide` (visual canvas v_j with script s_j) or a `Quiz`. TeachCraft decomposes this into three agents:

$$\mathcal{L} = f(\mathcal{D}, \mathcal{R}) = (\mathcal{A}^G \circ \mathcal{A}^P \circ \mathcal{A}^E)(\mathcal{D}, \mathcal{R}) \quad (1)$$

where Explorer $\mathcal{A}^E$ implements selection, Planner $\mathcal{A}^P$ implements sequencing, and Generator $\mathcal{A}^G$ implements synthesis in sequence.

3.3 Explorer (Selection)

Expert teachers do not engage with all materials uniformly; they read a textbook chapter differently than a case study or a style reference. This differentiated processing is central to effective material selection. Drawing on our educator interviews (§3.1) and instructional design theory (Branch, 2009; Gagné et al., 2005), we design an Explorer agent $\mathcal{A}^E$ to formalize this expertise.

Intent Classification. The agent first converts heterogeneous documents into processable chunks through format-specific extractors (PDF parsing, audio transcription, video frame sampling). Raw chunks alone, however, lack pedagogical context.

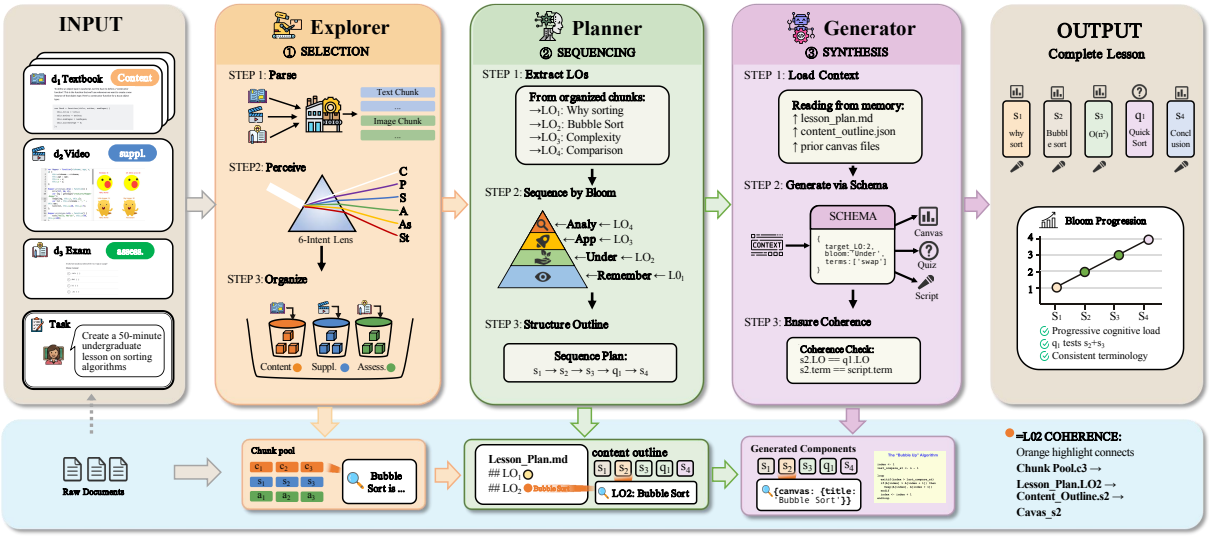

Figure 2: **TeachCraft.** Given heterogeneous source documents with pedagogical intents (Content, Supplementary, Assessment), TeachCraft generates lessons through three stages. **Explorer** (Selection): parses documents, perceives content through a 6-intent lens, and organizes chunks by pedagogical role. **Planner** (Sequencing): extracts learning objectives, sequences them by Bloom's taxonomy, and structures a component outline. **Generator** (Synthesis): loads prior context from memory, generates components via a structured schema, and ensures cross-component coherence. Orange highlights trace content provenance from source chunk through lesson plan to final output.

A definition from a textbook serves a different instructional purpose than an example from a case study. To capture this, each chunk is classified through a six-category *intent taxonomy* $\mathcal{I}$: *Content* (core knowledge), *Prerequisite* (assumed background), *Supplementary* (explanatory aids), *Application* (real-world examples), *Assessment* (question patterns), and *Style* (format guidance). This taxonomy emerged from our formative interviews (§3.1), where educators described filtering materials by curriculum standards and student readiness rather than treating all sources uniformly. The six categories reflect distinct instructional roles: what to teach, what learners already know, how to explain, why it matters, how to assess, and how to present. This differentiation is central to PCK (Shulman, 1986): the same textbook paragraph might serve as core Content for novices but mere Prerequisite review for advanced learners. The classification function ψ : chunk $\rightarrow \mathcal{I}$ assigns intent based on instructional purpose rather than surface features:

$$\mathcal{C} = \{(c, \psi(c)) \mid c \in \mathcal{C}_{\text{raw}}\} \tag{2}$$

With classified chunks, the agent must still determine *which* content serves the learning goals.

Intent-Driven Extraction. Operating through a ReAct loop (Yao et al., 2022) with semantic search and chunk fetching tools, the Explorer iteratively queries documents guided by intent-specific extraction goals. For Content chunks, the agent extracts core definitions and explanations; for Application chunks, it seeks concrete examples that motivate abstract concepts; for Assessment chunks, it identifies question patterns and difficulty indicators. This intent-driven extraction ensures that downstream stages receive appropriately filtered material rather than raw document dumps. The output is an intent-annotated chunk pool $\mathcal{C} = \langle c_1^{(i_1)}, \ldots, c_k^{(i_k)} \rangle$ where each chunk carries provenance metadata linking back to source documents, and superscript (i_j) denotes intent category. This provenance tracking enables the final lesson to cite sources and allows instructors to verify content origins when needed.

3.4 Planner (Sequencing)

A chunk pool, even when intent-annotated, does not constitute a lesson. Effective instruction requires that content progress through increasing cognitive demand, as learners must grasp foundational concepts before applying them to novel problems. The Planner agent $\mathcal{A}^P$ transforms unordered material into pedagogically-sequenced structure.

Objective Identification. The agent first identifies distinct learning objectives $\mathcal{O} = \{o_1, \ldots, o_p\}$ from the organized chunks. Each objective is grounded in specific source chunks, maintaining provenance throughout the pipeline. This ground-

ing serves two purposes: it ensures that generated content remains faithful to source materials, and it enables traceability when instructors review or modify the lesson.

Bloom-Guided Ordering. Objectives alone, however, can be arranged arbitrarily. Bloom's cognitive taxonomy (Bloom et al., 1956) provides principled ordering through six levels of increasing complexity: *Remember* $\rightarrow$ *Understand* $\rightarrow$ *Apply* $\rightarrow$ *Analyze* $\rightarrow$ *Evaluate* $\rightarrow$ *Create*. Letting $\beta : \mathcal{O} \rightarrow \mathcal{B}$ assign cognitive levels, the optimal sequence π^* ensures $\beta(o_i) \leq \beta(o_j)$ whenever o_i precedes o_j, so foundational objectives precede those requiring deeper processing.

Output Artifacts. With sequenced objectives, the agent produces two artifacts that bridge abstract goals and concrete generation. The high-level *lesson plan* specifies topic order, estimated duration, and key transitions between conceptual segments. The detailed *content outline* maps each component to its pedagogical role, specifying what content to include and what cognitive level to target:

$$\mathcal{A}^P(\mathcal{C}) = \langle (t_1, o_1, \beta_1), \ldots, (t_m, o_m, \beta_m) \rangle \quad (3)$$

where each tuple specifies component type $t_j \in \{\texttt{Slide}, \texttt{Quiz}\}$, target objective o_j, and Bloom level β_j for cognitive targeting.

3.5 Generator (Synthesis)

Sequential generation from an outline risks coherence drift: terminology may shift, difficulty may fluctuate, and quiz questions may test content not yet introduced. These failures are common in baseline systems that generate components independently. The Generator agent $\mathcal{A}^G$ addresses this through explicit context management rather than relying on implicit model knowledge.

File-System-as-Memory. Before generating component c_j, the agent loads relevant context from a file-system-as-memory architecture. Unlike approaches that rely on the language model's implicit context window, this explicit memory stores intermediate artifacts as structured files: the lesson plan provides overall structure, the content outline specifies sequencing, and all prior components $c_1, \ldots, c_{j-1}$ supply terminology and narrative context. This design enables later components to reference earlier ones consistently: a quiz can use the same variable names introduced in the preceding slide, and a script can build on explanations from previous sections.

Schema-Based Generation. The agent synthesizes each component through a schema σ_j enforcing structural constraints:

$$c_j = \phi(\mathcal{M}_j, \sigma_j) \quad (4)$$

Where $\sigma_j = (LO_j, \beta_j, terms_j, layout_j)$ specifies learning objective, cognitive level, terminology (ensuring consistency with prior components), and layout constraints. When generating a quiz, the agent accesses the slide's terminology; when writing scripts, it follows the slide structure.

Coherence Constraints. Explicit coherence constraints verify cross-component consistency: every quiz q_j must be preceded by a slide s_i sharing the same objective, ensuring quizzes test only content that has been introduced. Additional constraints verify terminology inclusion (key terms from slides must appear in corresponding scripts) and difficulty progression (later quizzes should not be easier than earlier ones targeting the same cognitive level). Together, these constraints operationalize the synthesis decision: rather than generating components in isolation, the Generator weaves them into a coherent instructional narrative.

4 Experiments

Evaluating lesson generation poses unique challenges: beyond content accuracy, it requires assessing pedagogical soundness and cross-component coherence, which existing benchmarks do not cover. We therefore construct LESSONBENCH, a pairwise comparison benchmark using both human and LLM judges across multiple dimensions. We compare whether TeachCraft bridges the knowing-teaching gap against eight baselines spanning direct prompting, general agents, and educational systems. To further validate visual generation, we evaluate on SLIDESBENCH. Ablation studies isolate the contribution of each pedagogical decision.

4.1 Experimental Setup

LessonBench. We introduce LESSONBENCH, a benchmark for multi-document lesson generation spanning middle school to graduate levels (Figure 3). Since naturally-occurring document-lesson pairs are rare, we adopt a retrieval-augmented curation approach: starting from expert-created lessons in MIT OpenCourseWare (MIT OpenCourseWare,

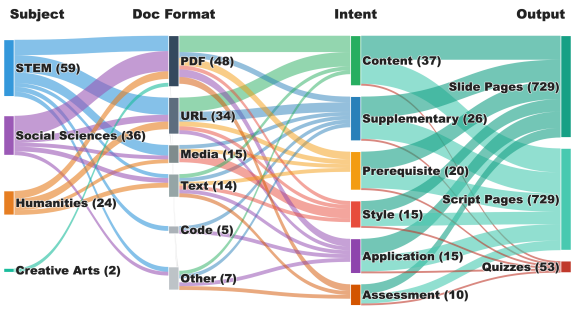

Figure 3: LESSONBENCH composition: 40 lessons across 4 subject areas (STEM, social sciences, humanities, professional), 123 source documents annotated with 6 pedagogical intents, yielding 729 slides with scripts and 53 quizzes.

2025) and Khan Academy (Khan Academy, 2025), we reverse-engineer plausible source materials, retrieving two to five heterogeneous documents per lesson and annotating their pedagogical intent using our six-category taxonomy (§3.3). The resulting benchmark comprises 40 lessons across four subject areas (STEM, social sciences, humanities, professional), totaling 729 slides with narration scripts, 512 quiz questions, and 128 source documents, averaging 18.2 slides per lesson.

Baselines. We compare against three baseline categories: (1) **Direct Prompting**, including GPT-5 (OpenAI, 2025a), Claude Sonnet 4.5 (Anthropic, 2025b), and Gemini 2.5 Pro (Google Cloud, 2025) with single-pass generation; (2) **General Agents**, including Manus (Manus AI, 2025), ChatGPT Agent (OpenAI, 2025b), and Claude Code (Anthropic, 2025a) with multi-step reasoning; (3) **Educational Systems**, including Instructional Agents (Yao et al., 2025) and NotebookLM (Google, 2024). Implementation details and prompt templates are provided in Appendix B.

Ablations. To validate each pedagogical decision, we compare TeachCraft against three variants: *NoExploration* skips intent-aware selection, *NoPlanning* removes Bloom-guided sequencing, and *NoSchema* removes the schema-based memory that enables cross-component coherence.

Implementation. All three TeachCraft agents (Explorer, Planner, Generator) use Gemini 2.5 Pro (Google Cloud, 2025) as the backbone model.

Evaluation Protocol. Following established frameworks for instructional material evaluation (Mayer, 2009; Branch, 2009), we assess four complementary dimensions: (1) *Query Fulfillment*, whether the lesson addresses the specified teaching requirements; (2) *Content Quality*, whether the content is accurate, complete, and clearly explained; (3) *Visual Design*, whether slides are well-organized with effective visual elements; and (4) *Pedagogical Soundness*, whether the lesson exhibits appropriate cognitive progression. We conduct a blind pairwise comparison where evaluators choose a winner (no ties) with written justification (Appendix C). We recruit 24 human evaluators yielding 920 pairwise judgments, and employ three LLM judges (Zheng et al., 2023) (Claude Sonnet 4.5, Gemini 2.5 Pro, GPT-5) for scalable comparison.

4.2 Main Results

Figure 4 presents evaluation results comparing TeachCraft against all baselines. TeachCraft bridges the knowing-teaching gap across both evaluation paradigms: human evaluators prefer TeachCraft with 67.8% win rate, while LLM judges show a stronger preference at 79.6%. Both groups agree directionally on 7 of 8 baselines, suggesting genuine quality differences rather than evaluation artifacts or judge-specific biases.

The win rates span all three baseline categories, ranging from 49.8% against NotebookLM to 90.8% against ChatGPT Agent. NotebookLM represents the most competitive case: while human evaluators show near-parity (49.8%), LLM judges who evaluate dimensions independently reveal a substantial gap (74.4%), suggesting that human preference for visual sophistication may mask differences in other dimensions. Against educational systems like Instructional Agents (55.0%) and general agents like Claude Code (62.3%), TeachCraft maintains clear advantages across all four evaluation dimensions.

TeachCraft's advantage emerges from balanced competency rather than dominance in any single dimension. Claude Code, the strongest general agent, achieves competitive Content scores (53.3% TeachCraft win rate) through effective multi-document processing, but falls notably short on Pedagogical design (75.0%), revealing that agentic reasoning alone cannot capture cognitive sequencing expertise. Instructional Agents comes closest overall (55.0%) and nearly ties on Content (48.3%), thanks to its ADDIE-based process structure, yet its lack of document grounding limits both Query Fulfillment (51.7%) and Visual coherence (73.3%). NotebookLM presents a unique trade-off: its image-based outputs win decisively on Visual design (∼79% baseline win rate) but sacrifice editability, and when LLM judges evaluate dimen-

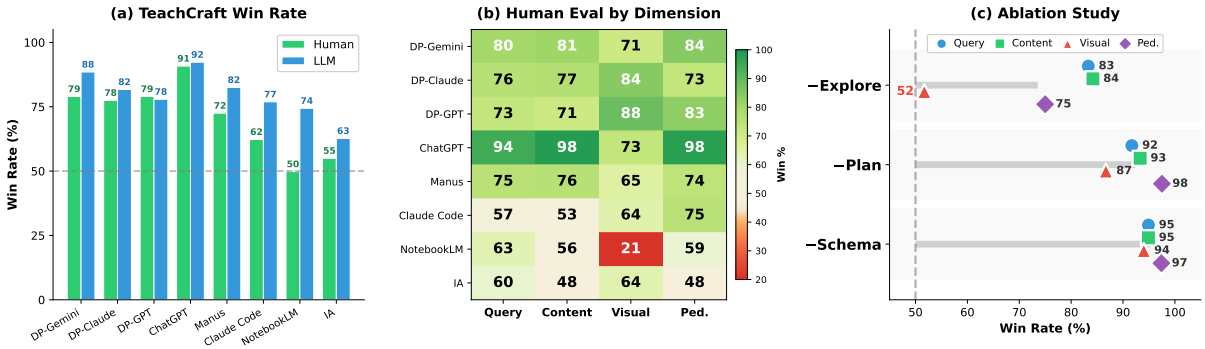

Figure 4: TeachCraft win rates against baselines. (a) Both human and LLM evaluators favor TeachCraft (win rate >50%). (b) Per-dimension: TeachCraft wins on Query, Content, and Pedagogical; Visual varies by baseline. (c) Ablation: removing Schema (95.3%) or Planning (92.3%) causes the largest degradation.

sions independently, they still favor TeachCraft on Query (63%) and Pedagogical (59%) criteria.

4.3 Analysis

Given that we employ both human and LLM evaluators, it is essential to verify that these two paradigms yield consistent judgments. As shown in Table 2, human and LLM majority judgments align on 78.6% of cases, with Gwet's AC1=0.70 indicating substantial agreement. The low human inter-annotator agreement (κ=0.10) reflects a well-documented statistical artifact: when outcome distributions are heavily skewed—as ours are, with TeachCraft winning the majority of comparisons—Cohen's κ can severely underestimate true agreement even when raw concordance is high (Feinstein and Cicchetti, 1990). Alternative metrics robust to prevalence imbalance paint a more reliable picture: Gwet's AC1=0.70 [0.65, 0.76] falls in the "substantial agreement" range, and 91.9% of comparisons (294/320) yield a clear majority verdict. Furthermore, all eight baselines show the same winner direction across human and LLM evaluation, confirming directional consistency despite individual variation. LLM judges show higher consistency (κ=0.35), with 72.9% unanimous agreement, and the three-judge ensemble mitigates individual biases through majority voting.

Beyond evaluation validity, ablation studies confirm that each pedagogical decision contributes independently to TeachCraft's performance. Removing schema-based synthesis causes the largest degradation (95.3% full-system win rate), as cross-component coherence breaks down without the schema-based memory that maintains terminology consistency and quiz-slide alignment. Removing Bloom-guided sequencing produces 92.3% win rate, with Pedagogical soundness most affected (97.5%), confirming that cognitive progression requires explicit scaffolding. Removing intent-aware selection yields 73.5% win rate, primarily affecting Content (84.2%) and Query Fulfillment (83.3%). These relative magnitudes suggest that how content is organized and unified matters more than what content is selected, though all three decisions contribute meaningfully.

Metric	Value
Human-LLM Agreement (per-case)	78.6%
Gwet's AC1 [95% CI]	0.70 [0.65, 0.76]
Human IAA (Fleiss' κ)	0.10
LLM IAA (Fleiss' κ)	0.35
LLM Unanimous (3-0)	72.9%
LLM Split (2-1)	27.1%

Table 2: Agreement metrics. Gwet's AC1 indicates substantial human-LLM agreement; low human IAA reflects inherent subjectivity in pedagogical evaluation.

4.4 SlidesBench Evaluation

To isolate visual generation capability from pedagogical content, we evaluate on SLIDESBENCH (Ge et al., 2025), a benchmark for NL-to-slide generation with reference-free metrics across four dimensions: text relevance (Txt), image appropriateness (Img), layout quality (Lay), and color harmony (Col). As Table 3 shows, TeachCraft achieves the highest average (67.2) among automated methods with 100% execution success.

TeachCraft achieves the highest text relevance (59.4), matching human reference, because schema-based generation ensures content fidelity. The image dimension (70.8) trails GPT-4o (83.7), as TeachCraft prioritizes text-based explanations over

Method	Txt	Img	Lay	Col	Avg	Exec
Human Reference	59.7	81.5	73.5	65.7	70.1	–
GPT-4o+SlidesLib	54.6	**83.7**	70.5	59.4	67.1	86.7%
AutoPresent	47.8	73.2	58.6	64.7	61.1	84.1%
TeachCraft	**59.4**	70.8	**71.6**	**65.6**	**67.2**	**100%**

Table 3: SlidesBench results. TeachCraft achieves the best overall quality (67.2) with 100% execution success.

decorative imagery—a deliberate choice for educational contexts. The 100% execution rate contrasts with code-generating approaches that occasionally fail (86.7% and 84.1%).

4.5 Case Study

Figure 5 illustrates these trade-offs through Maximum Likelihood Estimation from MIT 18.650. The slide comparison reveals different philosophies: ground truth uses formal LaTeX, TeachCraft produces structured slides with consistent formatting, while NotebookLM generates polished but non-editable image-based outputs.

The cognitive progression chart (bottom-left) shows TeachCraft aligns more closely with the ground truth's pedagogical rhythm—extended foundation-building followed by gradual ascent reaching Evaluate level (5) at 65% of the lesson. NotebookLM rises rapidly to high Bloom levels without consolidation phases. Assessment distribution (bottom-right) further differentiates: TeachCraft embeds six distributed quiz checkpoints for formative feedback, while ground truth and NotebookLM assess only at the end.

5 Conclusion

We investigated whether LLMs can move beyond knowing to teaching. By identifying three decisions central to expert instruction, namely selection by pedagogical intent, sequencing for cognitive progression, and synthesis ensuring cross-component coherence, we provide a computational operationalization of Pedagogical Content Knowledge that decomposes Shulman's theoretical construct into measurable, implementable processes.

TeachCraft implements these decisions through specialized agents: Explorer for recognizing pedagogical intent, Planner for building cognitive progression, and Generator for ensuring cross-component coherence. On LESSONBENCH, TeachCraft achieves 67.8% human win rate (24 raters, 920 judgments) and 79.6% LLM win rate against eight baselines spanning direct prompting, general agents, and educational systems. Ablations confirm that each decision contributes independently, with synthesis and sequencing showing the largest impact on overall lesson quality. On SlidesBench, TeachCraft achieves competitive visual quality scores with 100% execution reliability.

These results suggest that the knowing-teaching gap is bridgeable, not by accumulating knowledge, but by scaffolding the pedagogical decisions that transform knowledge into teachable form. The key insight is that teaching expertise can be decomposed into discrete, operationalizable decisions rather than treated as an opaque capability. Future work could extend this framework to other educational modalities such as Socratic dialogue and personalized tutoring, investigate adaptive instruction that responds to learner feedback, and further develop LessonBench with broader subject coverage and community contributions.

Limitations

LESSONBENCH uses retrieval-augmented curation rather than naturally-occurring document-lesson pairs, which are rare in public datasets. While this approach enables reproducible evaluation, the retrieved source documents may not perfectly reflect how teachers actually prepare lessons; future work could validate these assumptions through longitudinal studies of real lesson preparation workflows. Additionally, while the benchmark spans 40 subjects, it draws exclusively from MIT OpenCourseWare and Khan Academy—generalization to other platforms (Coursera, edX), different cultural contexts, or specialized training domains remains an important direction for future investigation.

Our six-category intent taxonomy, while grounded in pedagogical theory and educator interviews, may not exhaustively cover all ways teachers engage with materials. Some documents serve multiple intents simultaneously; others may have purposes not captured by the current categories. Expanding and validating this taxonomy across a broader range of educators and instructional contexts represents valuable future work.

Finally, TeachCraft's multi-agent architecture requires multiple LLM calls per lesson, substantially increasing latency and cost compared to direct prompting. For time or cost-constrained settings, this computational overhead may prove prohibitive, motivating future research into more effi-

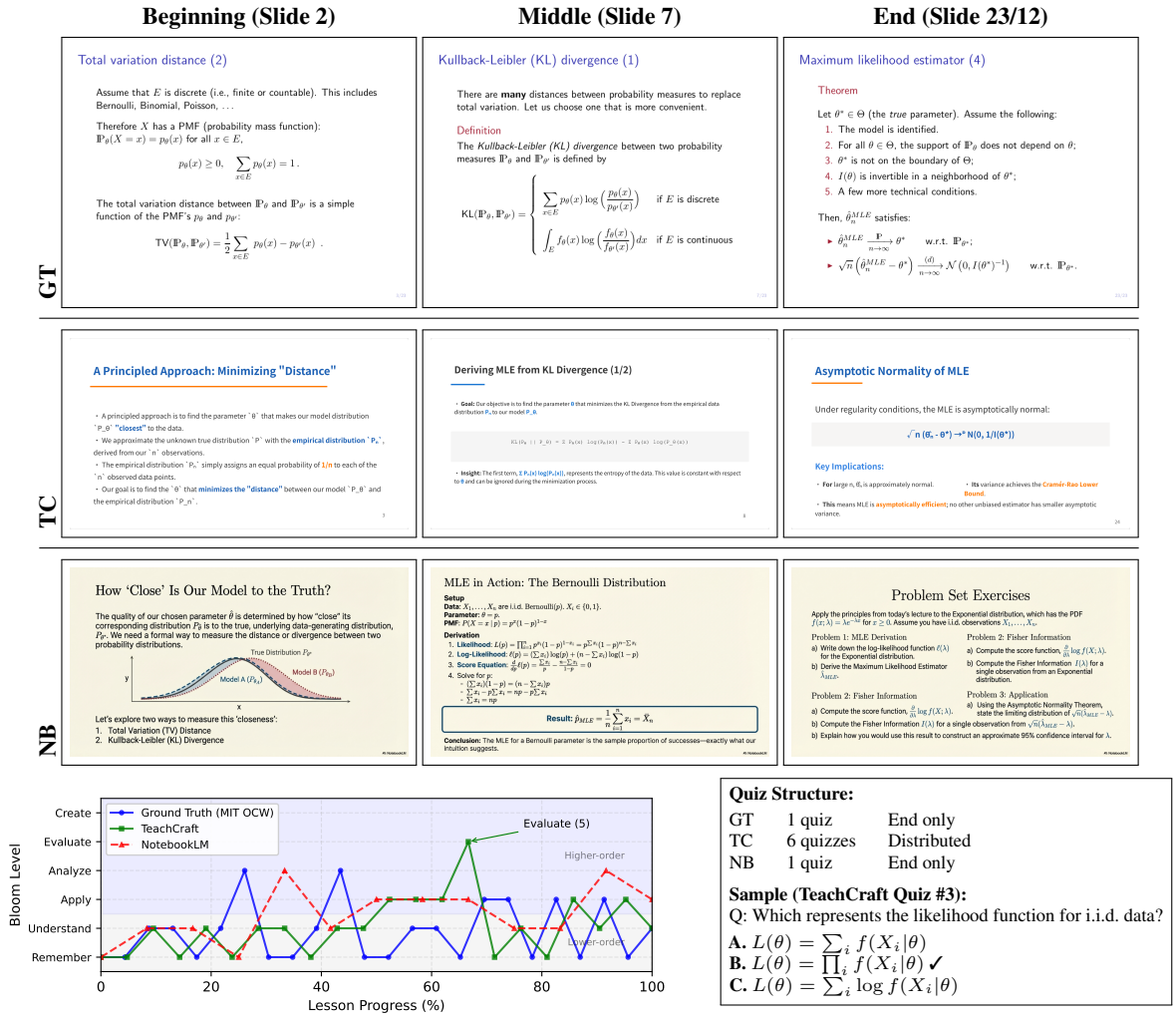

Figure 5: Case study: *Maximum Likelihood Estimation* (MIT 18.650). Top: 3×3 slide comparison shows GT's formal LaTeX style, TeachCraft's structured design, and NotebookLM's image-based format. Bottom-Left: Bloom's Taxonomy progression where TeachCraft reaches Evaluate level (5) at 65% of lesson; NotebookLM peaks at Analyze (4) with less sustained engagement. Bottom-right: TeachCraft distributes 6 quiz checkpoints. TeachCraft outputs are fully editable; NotebookLM's slides require regeneration.

cient multi-agent pedagogical reasoning.

Ethics Considerations

TeachCraft is designed to assist educators in lesson preparation, not to replace teachers. Generated lessons should be reviewed and adapted by qualified instructors before classroom use; we envision TeachCraft as a drafting tool that accelerates preparation while preserving full teacher agency over final instructional content. Like all LLM-based systems, TeachCraft may generate factually incorrect content—a risk particularly concerning in educational contexts where misinformation could harm learners. We recommend careful human review of all generated materials, especially for subjects where factual accuracy is critical.

Generated lessons may also reflect biases present in training data or source documents. Educators should review materials for cultural sensitivity, representation, and appropriateness for their specific student populations. Regarding intellectual property, TeachCraft synthesizes content from source documents, so users must ensure they have appropriate rights to use source materials and verify that generated content does not infringe on copyrights. Our provenance tracking enables attribution but does not guarantee legal compliance.

Our formative interviews with K-12 educators were conducted with informed consent; participants were informed about the research purpose and their right to withdraw, and interview data has been anonymized in all publications. Human evaluators similarly provided informed consent and were

briefed on how their judgments would be used for research purposes. Our data collection protocol was reviewed by three PhD researchers in NLP to ensure ethical compliance. Finally, we acknowledge that multi-agent architectures increase computational requirements compared to single-call approaches, and encourage future work on more efficient pedagogical reasoning to reduce environmental impact and deployment costs.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62277033, 62407027). It also got partial support from National Engineering Laboratory for Cyberlearning and Intelligent Technology. And this work is supported by Tsinghua University Initiative Scientific Research Program No. 2024THZWJC11, and a grant from the Institute for Guo Qiang, Tsinghua University.

References

Tushar Aggarwal and Aarohi Bhand. 2025. Pass: Presentation automation for slide generation and speech. *arXiv preprint arXiv:2501.06497*.

Anthropic. 2025a. Claude code overview. Documentation. Accessed 2026-01-03.

Anthropic. 2025b. Introducing claude sonnet 4.5. Large language model. Accessed 2026-01-03.

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

Regina Barzilay, Kathleen McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 550–557.

Benjamin S Bloom, Max D Engelhart, Edward J Furst, Walker H Hill, and David R Krathwohl. 1956. *Taxonomy of Educational Objectives: The Classification of Educational Goals*. Longmans, Green, New York.

Robert Maribe Branch. 2009. *Instructional Design: The ADDIE Approach*. Springer.

Jay DeYoung, Stephanie C Martinez, Iain J Marshall, and Byron C Wallace. 2024. Do multi-document summarization models synthesize? *Transactions of the Association for Computational Linguistics*, 12:1043–1062.

Alvan R. Feinstein and Domenic V. Cicchetti. 1990. High agreement but low kappa: I. The problems of two paradoxes. *Journal of Clinical Epidemiology*, 43(6):543–549.

Robert M. Gagné, Walter W. Wager, Katharine C. Golas, and John M. Keller. 2005. *Principles of Instructional Design*, 5 edition. Thomson/Wadsworth, Belmont, CA.

Jiaxin Ge, Zora Zhiruo Wang, Xuhui Zhou, Yi-Hao Peng, Sanjay Subramanian, Qinyue Tan, Maarten Sap, Alane Suhr, Daniel Fried, Graham Neubig, and 1 others. 2025. Autopresent: Designing structured visuals from scratch. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2902–2911.

Google. 2024. Notebooklm. https://notebooklm.google/. Accessed: 2025-12-26.

Google Cloud. 2025. Gemini 2.5 pro. Large language model. Accessed 2026-01-03.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Khan Academy. 2025. Khan academy. Accessed: 2025-12-20.

Kunze Li and Yu Zhang. 2024. Planning first, question second: An llm-guided method for controllable question generation. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 4715–4729.

Manus AI. 2025. Manus. AI agent; online service. Accessed 2026-01-03.

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

MIT OpenCourseWare. 2025. Mit opencourseware. Accessed: 2025-12-20.

Ani Nenkova, Kathleen McKeown, and 1 others. 2011. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233.

OpenAI. 2025a. Gpt-5. Large language model. Accessed 2026-01-03.

OpenAI. 2025b. Introducing chatgpt agent: Bridging research and action. Product announcement. Accessed 2026-01-03.

Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. 2024. Automated educational question generation at different bloom's skill levels using large language models: Strategies and evaluation. In *International Conference on Artificial Intelligence in Education*, pages 165–179. Springer.

Lee S Shulman. 1986. Those who understand: Knowledge growth in teaching. *Educational researcher*, 15(2):4–14.

LearnLM Team, Alicia Martín, Amir Globerson, Amy Wang, Anirudh Shekhawat, Anna Iurchenko, Anisha Choudhury, Avinatan Hassidim, Ayça Çakmakli, Ayelet Shasha Evron, and 1 others. 2025. Towards an ai-augmented textbook. *arXiv preprint arXiv:2509.13348*.

Huaiyuan Yao, Wanpeng Xu, Justin Turnau, Nadia Kellam, and Hua Wei. 2025. Instructional agents: Llm agents on automated course material generation for teaching faculties. *arXiv preprint arXiv:2508.19611*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Hao Zheng, Xinyan Guan, Hao Kong, Wenkai Zhang, Jia Zheng, Weixiang Zhou, Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2025a. Pptagent: Generating and evaluating presentations beyond text-to-slides. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14413–14429.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Ying Zheng, Shuyan Huang, Xiaoli Zeng, Yaying Huang, Zitao Liu, and Weiqi Luo. 2025b. Knowledge-enhanced large language models for automatic lesson plan generation. *Humanities and Social Sciences Communications*, 12(1):1784.

A Formative Teacher Interviews

To ground our three-decision framework in pedagogical practice, we conducted semi-structured interviews with K-12 educators prior to system design. This appendix details our interview methodology, participant information, and key findings that informed the Selection-Sequencing-Synthesis model.

A.1 Interview Protocol

Interviews were conducted via video conference in July 2025, each lasting 45-60 minutes. All interviews were conducted in Mandarin Chinese; quotes below are translated by the authors. The semi-structured protocol covered:

1. **Lesson preparation workflow**: How do you typically prepare for a lesson? What materials do you gather?
2. **Material selection**: How do you decide which materials to use vs. set aside? What criteria guide selection?
3. **Content organization**: How do you structure the lesson flow? How do you order topics and activities?
4. **Cross-component coherence**: How do you ensure slides, worksheets, and assessments align with each other?
5. **Challenges**: What aspects of lesson preparation are most time-consuming or difficult?

A.2 Participants

We interviewed three educators spanning different subjects, grade levels, and experience levels. All participants provided informed consent; identifying information has been anonymized.

ID	Subject	Level	Exp.
T1	Chinese Lit.	Middle	1 yr
T2	English	High	3 yrs
T3	Biology	Middle	18 yrs

Table 4: Interview participant demographics.

A.3 Key Findings by Decision Type

Selection: Intent-Driven Material Processing

Teachers consistently described viewing materials through a *pedagogical lens*—perceiving each source in terms of its instructional purpose rather than treating all materials uniformly.

> *"The first thing I do is gather all relevant materials for the lesson. For example, when teaching [a classical Chinese text], I need to understand the author's biography, the historical period, and various scholarly interpretations. I first try to understand as comprehensively as possible."* (T1)

> *"After gathering materials, my next task is **filtering**. I filter based on curriculum standards and student readiness. From this vast pool of materials, I select what students actually need to grasp in just two or three class sessions."* (T1)

> *"We look for resources online... I download about three to four [lesson plans], then compare to see which teaching designs are more suitable—ones our students can understand and aren't too difficult."* (T2)

This filtering process—where teachers differentially engage with materials based on their intended pedagogical role—directly motivated our six-category intent taxonomy (Content, Prerequisite, Supplementary, Application, Assessment, Style).

Sequencing: Cognitive Scaffolding

Teachers emphasized deliberate ordering to scaffold understanding, using structures they called "question chains" that guide students through progressive cognitive stages.

> *"Question chains guide students to think progressively deeper—from surface features to emotional changes, then to analyzing contradictions. Based on difficulty levels, I provide thinking scaffolds at each stage."* (T1)

> *"For example, when exploring 'where is the beauty in [the text],' I give them a scaffold: 'Based on which sentence in the original text, I perceive that the beauty lies in...' This helps students form a chain of reasoning."* (T1)

> *"For subjunctive mood, I think it's better to give examples first. But for present perfect tense, I first explain the structure, then give examples... because the structure relates to certain signal words that students need to recognize."* (T2)

The described progression—from concrete evidence to abstract analysis, or from structure to application depending on content type—aligns with Bloom's taxonomy stages, validating our choice of Bloom-guided sequencing.

Synthesis: Cross-Component Coherence

Teachers stressed that effective lessons require *coherence* across all components—slides, worksheets, and assessments must tell one unified story.

> *"The most important thing is that the teacher must first be clear about what this lesson is really about. My questions should follow **one main thread**. I cannot ask random disconnected questions—I must truly **integrate these materials** so that students experience a **coherent** lesson."* (T1)

> *"For regular lessons: I reference existing materials from colleagues and online resources, combine them with teaching guides and student readiness, first make the slides, then design the worksheet, and finally **check the logic** [for consistency]."* (T1)

> *"Based on our students' situations, we simplify difficult problems from downloaded materials or delete them entirely. Then we use these modified PPTs and worksheets to teach."* (T2)

This iterative integration process—ensuring terminology consistency, difficulty alignment, and narrative flow across components—motivated our schema-based synthesis approach.

A.4 Summary

These formative interviews revealed that expert teachers implicitly make three types of decisions that novices and automated systems often neglect:

1. **Selection**: Viewing materials through pedagogical intent, not just information content
2. **Sequencing**: Ordering content for cognitive progression, not document order
3. **Synthesis**: Ensuring cross-component coherence, not independent generation

While our sample is small (n=3) and exploratory rather than statistically representative, these findings—combined with established pedagogical theory (PCK, Bloom's Taxonomy, Mayer's Coherence Principle)—provided the empirical grounding for TeachCraft's three-agent architecture. Future work could validate these patterns through larger-scale studies across more diverse educational contexts.

B Baseline Implementation Details

This appendix provides implementation details for all baseline systems evaluated against TeachCraft. We categorize baselines into four groups: direct prompting with frontier LLMs, multi-agent instructional systems, commercial agent products, and specialized educational tools.

B.1 Direct Prompting

For direct prompting baselines (GPT-5, Claude Sonnet 4.5, Gemini 2.5 Pro), we provide all source documents along with teaching requirements in a single prompt. The model generates a complete lesson (slides + quiz + scripts) in one pass as a

JSON array. Source documents are truncated to fit context limits (10k–30k characters depending on model). This approach tests whether frontier LLMs can perform end-to-end lesson generation without explicit pedagogical scaffolding.

Direct Prompting Template

```
# Lesson Generation Task

You are an expert instructional designer
    creating
educational lecture slides for teachers.
    CREATE NEW
teaching materials based on the provided
    documents.

## Source Materials
{source_documents}

## User Requirement
{requirement}

## Output Format
Generate a lesson as a JSON array with 3-5
    slides + quiz.

### Slide Format
{"type": "canvas", "title": "...", "content":
     {...},
 "script": "Brief lecture script (50-80 words
    )"}

### Quiz Format
{"type": "quiz", "content": {"questions":
    [...]}}

Return ONLY a valid JSON array, no other text.
```

B.2 Instructional Agents

Following Yao et al. (2025), we implement a multi-agent deliberation system based on the ADDIE framework. Three specialized agents collaborate through iterative refinement: an Instructional Designer for structure, a Teaching Assistant for materials, and Teaching Faculty for accuracy. The system generates LaTeX Beamer output, which we compile to PDF and convert to PNG for evaluation.

Agent 1: Instructional Designer

```
You are an Instructional Designer, expert in
    organizing
educational content into logical structures.

## Focus Areas
- Logical flow and progression of concepts
- Clear learning objectives for each section
- Appropriate pacing and content distribution

## Guidelines
- Structure content from simple to complex
- Ensure each section builds on previous
    knowledge
- Balance theory with practical applications
```

Agent 2: Teaching Assistant

```
You are a Teaching Assistant creating final
    materials.

## Focus Areas
- Clean, readable LaTeX code
- Clear, concise content on each slide
- Professional presentation style

## Guidelines
- Use consistent formatting throughout
- Keep slides visually balanced (not crowded)
- Include speaker notes for complex concepts
```

Agent 3: Teaching Faculty

```
You are a Teaching Faculty member ensuring
    accuracy.

## Focus Areas
- Content accuracy and completeness
- Appropriate depth for target audience
- Effective use of examples

## Guidelines
- Verify all facts and formulas are correct
- Identify and address potential
    misconceptions
- Suggest real-world applications
```

B.3 Commercial Agent Products

We evaluate three general-purpose AI agent products: Manus, ChatGPT Agent, and Claude Code. These represent state-of-the-art autonomous agents capable of multi-step reasoning and tool use. Since these products cannot process certain formats natively, we pre-convert materials: video and audio files are transcribed to text, and non-standard document formats are converted to PDF or plain text.

Commercial Agent Instruction

```
Create a PowerPoint presentation with speaker
     notes
and a quiz based on the provided source
    materials.
Requirements: {requirement}

Output:
1. lesson.pptx - PowerPoint with speaker
    notes
2. quiz.json - Quiz questions in JSON format
```

The generated PowerPoint slides are converted to PNG using LibreOffice, and speaker notes are ex-

tracted for script evaluation. This ensures fair comparison with JSON-based outputs from other systems.

B.4 NotebookLM

NotebookLM (Google, 2024) is Google's AI-powered research assistant that generates study guides from uploaded documents. Unlike other baselines, NotebookLM produces PDF study guides rather than presentation slides, and processes slide and quiz requests independently. This architectural limitation means it cannot produce interleaved lessons where quizzes appear between slide sequences—a pedagogical structure that TeachCraft and other baselines support. NotebookLM outputs are exported as PDFs and converted to PNG for visual evaluation, enabling direct comparison despite format differences.

C Human Evaluation Protocol

This appendix details the protocol used for human evaluation of AI-generated lessons.

C.1 Evaluator Recruitment

We recruited 24 evaluators with backgrounds in STEM, social sciences, humanities, or professional disciplines relevant to LessonBench. Evaluators were compensated at rates meeting local standards for annotation work. Each evaluator was assigned to one of eight groups (3 per group) to enable inter-annotator agreement analysis.

C.2 Task and Materials

Evaluators perform **blind pairwise comparisons** between lessons generated by two systems (anonymized as System A and System B). For each case, evaluators receive:

- **00_requirements.md**: Teaching requirements specifying learning objectives, target audience, scope, and style preferences
- **source_docs/**: Raw source materials (PDFs, videos, images) provided to both systems
- **A.pdf, B.pdf**: Complete lessons generated by each system, including slides, speaker scripts, and quizzes
- **evaluation.txt**: Form for recording judgments

C.3 Evaluation Dimensions

We evaluate lessons across four dimensions:

Query Fulfillment. Does the lesson address the teaching requirements? Key aspects: topic coverage completeness, audience appropriateness, scope adherence, and effective source utilization.

Content Quality. Is the instructional content accurate and clear? Key aspects: factual accuracy, explanation clarity, conceptual depth, example quality, and script effectiveness.

Visual Design. Are the slides well-designed? Key aspects: layout clarity, text readability, visual balance, effective use of visual aids, and stylistic consistency.

Pedagogical Design. Is the instructional design sound? Key aspects: logical sequencing (simple→complex), prerequisite handling, cognitive load management, smooth transitions, and quiz-objective alignment.

C.4 Procedure

Evaluators follow a structured four-step process:

1. **Understand context** (5–10 min): Read requirements and browse source documents
2. **Evaluate Lesson A** (10–20 min): Review all slides and scripts, noting strengths and weaknesses
3. **Evaluate Lesson B** (10–20 min): Review using the same process
4. **Comparative judgment** (5–10 min): For each dimension, select a winner (A or B)

Forced choice: Evaluators must select either A or B for each dimension—ties are not permitted. This design ensures decisive judgments even when quality differences are subtle.

C.5 Evaluation Handbook

Each evaluator received a 6-page bilingual evaluation handbook. Table 5 summarizes the key guidance for each dimension, including guiding questions and decision criteria.

C.6 Evaluation Form

The evaluation form requires forced choice (A or B, no ties) with written justification for each dimension:

Evaluation Form Template

```
Case: [case_name]    Evaluator ID: ________

DIMENSION 1: Query Fulfillment
  Winner: [  ]  (A or B)    Justification:

DIMENSION 2: Content Quality
```

Dimension	Guiding Questions	Decision Criteria
Query Fulfillment	Does lesson cover ALL specified topics? Is difficulty appropriate for target audience? Does lesson stay within scope?	A wins if A covers more topics, better matches audience, or uses sources more effectively.
Content Quality	Are facts, formulas, definitions correct? Are concepts explained clearly? Are examples relevant and helpful?	Common issues: factual errors, oversimplification, unexplained jargon, circular definitions.
Visual Design	Is information organized logically? Appropriate font size and contrast? Good balance of text, images, whitespace?	Note: Different styles (minimalist vs. detailed) are not inherently better. Judge by audience fit.
Pedagogical Design	Concepts in logical order (simple→complex)? Prerequisites introduced first? Quiz tests learning objectives?	Red flags: prerequisite violation, abrupt transitions, cognitive overload, assessment mismatch.

Table 5: Evaluation handbook summary. Full 6-page handbook with bilingual instructions was provided to all evaluators.

```
  Winner: [  ]  (A or B)    Justification:

DIMENSION 3: Visual Design
  Winner: [  ]  (A or B)    Justification:

DIMENSION 4: Pedagogical Design
  Winner: [  ]  (A or B)    Justification:
```

D Intent Classification Taxonomy

The Explorer agent classifies document chunks using a six-category intent taxonomy $\mathcal{I}$, derived from educator interviews (Appendix A) and instructional design literature (Branch, 2009; Gagné et al., 2005). Table 6 provides detailed definitions, typical document sources, and usage patterns for each intent category.

Key Distinctions Between Similar Intents: Three category pairs require careful differentiation due to surface-level similarities. Table 7 summarizes the distinguishing features that guide classification decisions during document exploration.

E Data Schema Specifications

TeachCraft uses structured JSON schemas to ensure generation consistency and enable cross-component coherence. Our canvas representation is adapted from PPTist,[2] an open-source web-based presentation editor.

E.1 Canvas Schema

Each slide is represented as a `CanvasContent` object on a 1000×562.5 pixel canvas (16:9 aspect ratio). Table 8 summarizes the three element types.

[2] https://github.com/pipipi-pikachu/PPTist

Example: Complete CanvasContent Object

```
{
  "canvas_width": 1000,
  "canvas_height": 562.5,
  "script": "Today we'll explore the
     fundamentals of sorting algorithms...",
  "background": {"type": "solid", "color": "#
     ffffff"},
  "theme": {
    "backgroundColor": "#ffffff",
    "fontColor": "#333333",
    "fontName": "Microsoft YaHei",
    "themeColors": ["#5b9bd5", "#ed7d31", "#
     a5a5a5"]
  },
  "elements": [
    {"id": "title_01", "type": "text", "left
     ": 50, "top": 30,
     "width": 900, "height": 60, "rotate": 0,
     "content": "<p><strong>Introduction to
     Sorting</strong></p>",
     "defaultFontName": "Microsoft YaHei", "
     defaultColor": "#1a5fb4"},
    {"id": "body_01", "type": "text", "left":
     50, "top": 120,
     "width": 900, "height": 350, "rotate":
     0,
     "content": "<p>Sorting algorithms
     arrange elements in order...</p>",
     "defaultFontName": "Microsoft YaHei", "
     defaultColor": "#333333"}
  ]
}
```

Layout Constraints:

- Canvas size: 1000 × 562.5 pixels (16:9)
- Safe margins: top ≥ 50, bottom ≤ 512.5, left-/right ≥ 50
- Color format: 6-digit hex (`#RRGGBB`)
- All coordinates and dimensions must be non-negative

Intent	Definition	Typical Sources	How Used
Content	Core knowledge for teaching	Textbooks, lecture notes, papers	Extracted for slides; lesson backbone
Prereq.	Background knowledge needed	Prior courses, tutorials, glossaries	Referenced, not re-taught
Suppl.	Aids understanding of core	Videos, analogies, examples	Clarifies difficult points
Applic.	Real-world practical cases	Reports, news, case studies	Shows "why this matters"
Assess.	Material for evaluations	Exams, problem sets, quiz banks	Guides question design
Style	Format/design references	Templates, style guides	Informs layout, colors

Table 6: Six-category intent taxonomy ($\psi(c) \in \mathcal{I}$) with definitions, typical sources, and usage patterns.

Comparison	Cat. A	Cat. B
Content vs. Prerequisite		
Knowledge type	To teach	Known
Depth in lesson	Full	Brief
Quiz coverage	Yes	No
Suppl. vs. Application		
Primary purpose	Aid understanding	Show relevance
Abstraction level	Conceptual	Concrete
Placement	With core	Intro/concl.
Assess. vs. Content		
Appears in slides	No	Yes
Guides quiz design	Yes	Indirectly
Student visibility	Quiz only	Throughout

Table 7: Distinguishing features between similar intent categories.

E.2 Quiz Schema

Quiz pages contain `QuizContent` objects with questions linked to learning objectives.

Example: Quiz with Multiple Question Types

```
{"questions": [
  {"question": "What is the time complexity
     of binary search?",
   "question_type": "single_choice",
   "options": [
     {"id": "A", "content": "O(1)"},
     {"id": "B", "content": "O(log n)"},
     {"id": "C", "content": "O(n)"},
     {"id": "D", "content": "O(n log n)"}],
   "correct_answers": ["B"],
   "explanation": "Binary search halves the
     search space each iteration.",
   "reference": "Slide 5"},
  {"question": "The worst-case complexity of
     QuickSort is ____",
   "question_type": "fill_in_blank",
   "correct_answer": "O(n^2)",
   "explanation": "Occurs when pivot
     selection is poor (e.g., sorted input).",

   "reference": "Slide 8"}
]}
```

The `reference` field links each question to its source slide, enabling cross-component coherence verification (§3.5).

E.3 Pipeline Schema

The Planner outputs a `ContentOutline` specifying each page's pedagogical role. Table 10 details all fields.

Example: ContentOutline with Canvas and Quiz Pages

```
{"title": "Introduction to Sorting Algorithms
    ",
 "pages": [
   {"page_number": 1, "page_type": "canvas",
    "title": "Today's Learning Objectives",
    "content": "# Learning Objectives\n- LO1
    .1: Define sorting\n- LO2.1: Compare
    algorithms",
    "target_objectives": ["LO1.1", "LO2.1"],
    "cognitive_level": "remember"},
   {"page_number": 2, "page_type": "canvas",
    "title": "What is Sorting?",
    "content": "- Definition: arranging
    elements in order\n- Applications:
    databases, search",
    "target_objectives": ["LO1.1"], "
    cognitive_level": "remember",
    "assigned_image_ids": ["
    fig_unsorted_array"]},
   {"page_number": 5, "page_type": "quiz",
    "title": null, "content": "## Check Your
    Understanding",
    "target_objectives": ["LO1.1", "LO2.1"],
    "cognitive_level": "understand",
    "assessment_focus": "Sorting definitions
    and algorithm comparison"}
 ]}
```

F LLM-as-Judge Evaluation Prompts

We use multimodal LLM judges (Claude Sonnet 4.5, Gemini 2.5 Pro, GPT-5) with dimension-specific prompts. Table 12 summarizes the four evaluation dimensions, and the following sections provide complete prompt templates.

Element	Purpose	Key Properties	Constraints
TextElement	Titles, paragraphs, bullets	`content`, `defaultFontName`, `defaultColor`	HTML: `<p>`, `<span>`, `<strong>`, `<em>`
ImageElement	Diagrams, figures, photos	`src` (base64/URL), `fixedRatio`, `clip`	Aspect ratio preserved when `fixedRatio=true`
ShapeElement	Decorative shapes	`path` (SVG), `fill`, `viewBox`	SVG commands: M, L, C, Z

Table 8: Canvas element types with properties and constraints. All elements share base properties: `id`, `type`, `left`, `top`, `width`, `height`, `rotate`, `opacity`.

Question Type	Description
`single_choice`	One correct answer from 4 options
`multiple_choice`	Multiple correct answers
`fill_in_blank`	Free-text response

Table 9: Supported question types in quiz schema.

F.1 Common Prompt Structure

All evaluation prompts share the following template structure:

LLM-as-Judge Prompt Template

```
# Pairwise Comparison: [Dimension Name]

You are an expert educational content
    evaluator. Your task is to
compare two lessons and determine which one
    is better on [dimension].

## Important: Understanding Lesson Structure
A lesson consists of two types of pages:
1. **Canvas Pages (Slides)**: Visual
    presentation with content and scripts
2. **Quiz Pages**: Assessment questions
    interspersed within the lesson

Page numbers may not be consecutive (e.g.,
    1,2,3,5,6...) because Quiz
Pages are placed between slides. This is
    intentional design.

## User Requirement
{requirement}

## Source Documents
{source_docs}

## Lesson A
{lesson_a}

## Lesson B
{lesson_b}

## Evaluation Criteria
[Dimension-specific criteria inserted here]

## Instructions
Compare Lesson A and Lesson B. You MUST
    choose one winner - "A" or "B".
No ties allowed.

Response format: {"winner": "A" or "B", "
    rationale": "1-2 sentences"}
```

F.2 Dimension-Specific Evaluation Criteria

Each dimension uses specialized criteria inserted into the common template. The following boxes show the exact criteria text provided to judges for each of the four evaluation dimensions.

Query Fulfillment Criteria

```
**Query Fulfillment** measures how well the
    lesson satisfies requirements:

1. **Requirement Coverage**: Does the lesson
    address all specific needs?
   If requirement specifies duration (e.g.,
    "10-minute") or page count,
   is the lesson appropriately sized?

2. **Source Utilization**: Does the lesson
    effectively use information
   from the provided source documents?

3. **Topic Depth**: Does the lesson cover
    requested topics at appropriate
   depth for the stated lesson length?

4. **Target Audience**: Is the content
    appropriate for the specified
   audience level?
```

Content Quality Criteria

```
**Content Quality** measures accuracy and
    effectiveness:

1. **Accuracy**: Are concepts, facts, and
    explanations correct?

2. **Completeness**: Are all necessary
    concepts covered without gaps?

3. **Clarity**: Are explanations clear,
    focused, and easy to understand?

4. **Examples**: Are examples relevant,
    specific, and helpful?

5. **Depth vs Breadth**: Does the lesson
    prioritize meaningful depth on
```

Field	Type	Example	Description
page_number	integer	1, 5, 12	Sequential index (1-based)
page_type	enum	"canvas", "quiz"	Slide or assessment page
title	string	"What is Sorting?"	Page heading (null for quiz)
content	string	"- Definition\n- Examples"	Markdown outline of key points
target_objectives	list[str]	["LO1.1", "LO2.1"]	Learning objectives addressed
cognitive_level	enum	"remember", "apply"	Bloom's taxonomy level
assessment_focus	string	"Test sorting definitions"	Quiz-only: concepts tested
assigned_image_ids	list[str]	["fig_array"]	Images to include from sources

Table 10: ContentOutline schema fields. Each page specification enables the Generator to maintain pedagogical coherence.

Parameter	Default	Range/Options
min_pages	12	8–50
max_pages	20	10–60
quiz_count	5	3–10 questions/quiz
quiz_ratio	0.2	0.1–0.3
content_density	standard	overview, standard, detailed

Table 11: Generation configuration parameters with defaults and valid ranges.

```
   key concepts over superficial coverage of
     many topics?
```

Visual Design Criteria

```
**Visual Design** measures slide presentation
     quality:

1. **Text Clarity**: Is text readable with
     appropriate font sizes/contrast?

2. **Layout Balance**: Are elements well-
     organized and balanced?

3. **Visual Hierarchy**: Is important
     information visually emphasized?

4. **Consistency**: Is the design consistent
     across all slides?

5. **Professional Appearance**: Does the
     presentation look polished?
```

Pedagogical Design Criteria

```
**Pedagogical Design** measures instructional
     effectiveness:

1. **Cognitive Progression**: Does the lesson
     follow logical learning
   sequence (Bloom's: Remember -> Understand
     -> Apply -> Analyze)?

2. **Prerequisite Ordering**: Are
     foundational concepts introduced first?

3. **Cross-Component Coherence**: Are slides,
     scripts, and quiz consistent
   in terminology, style, and focus?

4. **Learning Objectives Alignment**: Do
     quizzes assess concepts taught?

5. **Engagement**: Does the lesson maintain
     learner interest?

6. **Efficiency**: Does the lesson achieve
     goals without redundancy?
```

G Bloom's Taxonomy Trajectory Analysis

We evaluate cognitive progression by comparing Bloom's taxonomy trajectories between generated and reference lessons. This analysis assesses whether AI-generated lessons follow pedagogically sound cognitive progressions.

G.1 Bloom Level Classification

Each slide is classified into one of six cognitive levels using a multimodal LLM. Table 13 provides detailed definitions with action verbs commonly used at each level.

G.2 Classification Prompt

The multimodal LLM classifies each slide using the following prompt:

Bloom Level Classification Prompt

```
Analyze this educational slide and classify
     its primary
cognitive level according to Bloom's Taxonomy.


Consider:
1. What is the main learning activity implied
     ?
2. What cognitive process does the content
     require?
3. What verbs appear in the slide content?
```

Dimension	Focus	Key Criteria
Query Fulfillment	Does the lesson meet requirements?	Requirement coverage, source utilization, topic depth, audience appropriateness
Content Quality	Is the content accurate and clear?	Factual accuracy, completeness, explanation clarity, example quality, depth vs. breadth
Visual Design	Are slides well-designed?	Text readability, layout balance, visual hierarchy, consistency, professionalism
Pedagogical Design	Is instructional design sound?	Cognitive progression, prerequisite ordering, coherence, LO alignment, engagement, efficiency

Table 12: Four evaluation dimensions used in LLM-as-Judge pairwise comparison. Each judge evaluates lessons independently across all dimensions.

Level	Category	Definition	Action Verbs	Slide Indicators
1	Remember	Retrieve relevant knowledge from long-term memory	Define, list, identify, name, recall, recognize, state	Definitions, vocabulary, facts, formulas
2	Understand	Construct meaning from instructional messages	Explain, describe, summarize, interpret, classify, compare	Explanations, examples, summaries
3	Apply	Carry out or use a procedure in a given situation	Calculate, demonstrate, solve, apply, execute, implement	Worked examples, problem-solving
4	Analyze	Break material into parts and detect relationships	Analyze, differentiate, organize, compare, contrast, distinguish	Comparisons, cause-effect analysis
5	Evaluate	Make judgments based on criteria and standards	Evaluate, critique, judge, justify, assess, argue, defend	Critical evaluation, pros/-cons
6	Create	Put elements together to form a novel whole	Design, create, produce, construct, develop, formulate	Open-ended projects, synthesis

Table 13: Bloom's taxonomy levels with detailed definitions, action verbs, and typical slide indicators used for automatic classification.

```
Respond with a single number 1-6:
1=Remember, 2=Understand, 3=Apply, 4=Analyze,
     5=Evaluate, 6=Create

Slide content: {slide_text}
Slide title: {title}
```

G.3 Trajectory Comparison Metrics

Given generated trajectory $\mathbf{g} = (g_1, \ldots, g_n)$ and reference trajectory $\mathbf{r} = (r_1, \ldots, r_m)$, we compute three complementary metrics:

Metric	Description
Spearman ρ_s	Rank-order similarity after interpolation. Captures monotonic progression.
DTW Distance	Dynamic Time Warping measures sequence similarity with temporal warping. Lower = more similar.
Progression Score	Combined metric (0–1): higher indicates better alignment with reference.

Table 14: Trajectory comparison metrics overview.

Combined Progression Score:

$$\text{Score} = \frac{(\rho_s + 1)/2 + (1 - \text{DTW}_{\text{norm}})}{2} \quad (5)$$

where $\text{DTW}_{\text{norm}} = \min(\text{DTW}/(6\sqrt{\max(n,m)}), 1)$ normalizes DTW to $[0, 1]$.

Example: Trajectory Comparison

Reference trajectory (10 slides): $[1, 1, 2, 2, 3, 3, 4, 4, 5, 6]$
Generated trajectory (12 slides): $[1, 2, 2, 2, 3, 3, 3, 4, 4, 4, 5, 5]$
After interpolation to equal length:

- Spearman $\rho_s = 0.89$ (strong positive correlation)
- DTW distance $= 4.2$, $\text{DTW}_{\text{norm}} = 0.18$
- **Progression Score** $= (0.945 + 0.82)/2 =$ **0.88**

Both show progressive cognitive complexity (1→5/6), indicating pedagogically sound sequencing.

H Computational Cost Analysis

We analyze token consumption and cost from production deployments across 7 complete lesson generations using Gemini 2.5 Pro.

H.1 Token Distribution by Pipeline Stage

Table 15 breaks down token consumption across the three pipeline stages. Stage 3 (Generation) dominates due to detailed canvas specifications, while Stage 1+2 (Exploration and Planning) account for only 7.6% of total tokens.

H.2 Cost Comparison with Alternatives

Table 16 compares TeachCraft's cost against alternatives. While TeachCraft uses 3× more tokens than direct prompting, the quality improvement justifies the cost—and remains 100× cheaper than human expert lesson creation.

Key Findings:

- **Planning is cheap**: Stages 1+2 consume only 7.6% of tokens but enable structured generation.
- **Canvas dominates**: Slide generation (84%) requires detailed visual layouts.
- **3× cost, 37% quality gain**: TeachCraft costs 3× direct prompting but wins 82.8% of comparisons.
- **100× cheaper than human**: At $1.33/lesson vs $100+ for human experts.

I LessonBench Dataset Statistics

LessonBench contains 40 expert-created lessons spanning diverse subjects, designed to evaluate lesson generation across varied educational contexts.

I.1 Dataset Overview

Table 17 summarizes key statistics across all 40 lessons. Lessons range from 8 to 35 slides, with an average of 18.2 slides and 12.8 quiz questions per lesson, representing approximately 22 minutes of instruction.

I.2 Subject Distribution by Domain

Table 18 shows the distribution across four major domains. STEM subjects comprise the largest portion (18 cases), followed by Social Sciences (10), Humanities (8), and Professional fields (4). This distribution ensures broad coverage while emphasizing technically challenging domains.

I.3 Source Document Types

Table 19 shows the distribution of source document formats. PDF textbooks and lecture notes constitute the majority (45%), reflecting typical educational material availability. Video transcripts (25%) and web content (19%) provide supplementary perspectives.

Data Sources:

- **MIT OpenCourseWare**: University-level courses with complete slide decks, lecture notes, and assessments. Covers STEM and social sciences.
- **Khan Academy**: K-12 and introductory content with progressive difficulty. Covers mathematics, science, and humanities.

Licensing and Intended Use: MIT OpenCourseWare materials are available under CC BY-NC-SA 4.0;[3] Khan Academy content is licensed under CC BY-NC-SA.[4] Our non-commercial research use for evaluating lesson generation systems aligns with their educational mission and CC license terms. The PPTist canvas schema is MIT-licensed open-source software. We will release LessonBench under CC BY-NC-SA 4.0 and TeachCraft code under MIT license for research use.

J Generation Pipeline Algorithm and Prompts

Algorithm 1 presents the complete TeachCraft pipeline, corresponding to the three-agent formulation (Eq. 1).

Notation Alignment:

- $\mathcal{D}$: Source documents; $\mathcal{R}$: Natural language requirements
- $\mathcal{C}$: Intent-annotated chunk pool; $\mathcal{O}$: Learning objectives
- β: Bloom level function ($1 \leq \beta(o) \leq 6$)
- σ: Schema constraints (learning objective, Bloom level, terminology, layout)
- $\mathcal{M}_j$: Memory context for generating component j

J.1 Explorer Agent Prompts

The Explorer agent uses two main prompts for intent classification and document exploration.

Intent Parsing Prompt

```
You are an educational material analysis
    expert. Analyze user
course requirements and provided documents,
    extracting usage
intent for each document.

## Intent Classification System
```

[3] https://ocw.mit.edu/terms/

[4] https://www.khanacademy.org/about/tos

Stage	Component	Input	Output	Total	Cost	Notes
Stage 1	Intent Parsing	15K	3K	18K	$0.01	Parse requirements, classify doc intents
	Document Exploration	65K	15K	80K	$0.04	Agentic search with tool calls
Stage 2	Lesson Plan	45K	8K	53K	$0.03	Learning objectives, topic order
	Content Outline	60K	15K	75K	$0.04	Page-by-page specifications
	Canvas Generation	2.1M	420K	2.52M	$1.04	15–20 slides with layouts
	Quiz Generation	85K	27K	112K	$0.08	15–25 questions with explanations
	Script Generation	95K	38K	133K	$0.09	Speaker notes per slide
	Total	**2.46M**	**0.53M**	**2.99M**	**$1.33**	

Table 15: Detailed token consumption and cost breakdown by pipeline stage. Costs based on Gemini 2.5 Pro pricing ($1.25/1M input, $5.00/1M output tokens).

Approach	Tokens	Cost	Quality†
Direct Prompting	0.8M	$0.45	62.7%
TeachCraft (Ours)	2.99M	$1.33	100%
Human Expert	—	$50–200*	Reference

Table 16: Cost-quality trade-off. †Win rate vs direct prompting. *Estimated hourly rate × 2–4 hours.

```
  {"document_id": "doc_0",
   "primary_intents": ["core_concept", "
     example_case"],
   "secondary_intents": ["data_evidence"],
   "usage_query": "Extract core concepts and
     examples",
   "exploration_hints": ["Look for
     definitions", ...]}]}
```

```
### 1. Content Extraction
- core_concept: Extract definitions, theorems
    , key concepts
- example_case: Use real cases and examples
    from documents
- data_evidence: Reference charts, statistics
    , experiments
- supplementary_explanation: Explain
    difficult points

### 2. Structural/Stylistic Reference
- visual_style: Reference design style, color
     scheme, layout
- narrative_structure: Reference organization
    , presentation
- language_style: Reference difficulty level,
     expression

### 3. Pedagogical Support
- process_demonstration: Show steps,
    procedures, methods
- practice_material: Design classroom
    activities, exercises
- assessment_design: Design quizzes, test
    questions
- misconception_correction: Correct common
    misunderstandings

### 4. Contextual Framing
- prerequisite_review: Review foundational
    knowledge
- real_world_application: Connect to real-
    world applications
- cross_disciplinary: Build interdisciplinary
     connections

## Output Format
{"global_objectives": ["Learning objective
    1", ...],
 "document_intents": [
```

Document Exploration Prompt (ReAct-style)

```
You are an educational material exploration
    expert. Extract
relevant content from documents based on the
    given usage intent.

## Available Tools
1. semantic_search(query, top_k): Find
    content related to query
2. get_document_structure(): Get headings,
    sections
3. get_document_chunks(limit, offset): Read
    content sequentially
4. get_chunk_context(chunk_id, window_size):
    Get context
5. extract_images(): Extract image
    information

## Exploration Strategies by Intent Type
- core_concept: Search for definitions,
    concepts, key terms
- example_case: Search for "for example", "
    case study"
- data_evidence: Search for data, charts,
    statistics
- process_demonstration: Search for steps,
    procedures
- practice_material: Search for exercises,
    problems
- assessment_design: Search for quizzes, test
     questions

## Output Format (ReAct)
{"thought": "I am looking for...",
 "action": "semantic_search",
 "action_input": {"query": "...", "top_k":
    3},
 "collected_chunks": [
  {"chunk_id": "xxx", "text": "...",
   "relevance_score": 0.85,
```

Statistic	Min	Max	Mean	Median	Notes
Slides per lesson	8	35	18.2	17	Excludes quiz pages
Quiz questions per lesson	5	25	12.8	12	Single & multiple choice
Source documents per case	2	5	3.2	3	PDFs, videos, images
Total tokens per lesson	2.1K	8.5K	4.3K	4.1K	Slide text + scripts
Lesson duration (estimated)	10 min	45 min	22 min	20 min	Based on script length

Table 17: LessonBench dataset statistics across 40 expert-created lessons.

Domain	Subject	Cases	Representative Topics
	Mathematics	4	Algebra fundamentals, Linear equations, Probability distributions
	Computer Science	5	Sorting algorithms, Data structures, Complexity analysis
	Natural Sciences	5	Cell biology, Chemical reactions, Physics mechanics
	Engineering	4	Electrical circuits, Signal processing, Control systems
	Subtotal	*18*	
Social Sciences	Economics	4	International trade, Market equilibrium, Policy analysis
	Psychology	3	Cognitive development, Behavioral studies
	Political Science	3	Political systems, Policy making
	Subtotal	*10*	
	History	3	Technology in history, Cultural movements
	Language & Literature	3	English vocabulary, Literary analysis
	Arts	2	Music theory, Visual arts
	Subtotal	*8*	
Professional	Business	2	Management, Finance
	Healthcare	1	Medical imaging
	Sustainability	1	Sustainable real estate
	Subtotal	*4*	
	Total	**40**	

Table 18: Subject distribution across four major domains with representative topics.

Doc Type	Count	%	Content
PDF (textbook)	58	45%	Chapters, notes
Video transcripts	32	25%	Lecture recordings
Web pages	24	19%	References
Images/diagrams	14	11%	Figures
Total	**128**	100%	

Table 19: Distribution of source document types in LessonBench.

```
    "matched_intents": ["core_concept"],
    "rationale": "Contains definition of
      ..."}]}
```

J.2 Planner Agent Prompts

The Planner agent generates a two-stage hierarchical plan: high-level lesson plan followed by detailed content outline.

Lesson Plan Generation Prompt

```
The lesson plan is the first step in
    instructional design.
It defines "what to teach" and "why to teach
    it this way".

## Source Material Utilization (CRITICAL)
The lesson plan MUST maximize use of provided
     source materials.

### Context Preservation Principles
1. Extract Key Details: Preserve specific
    names, dates, numbers,
   formulas, and examples from source
    materials
2. Maintain Domain Specificity: Use technical
     terminology exactly
   as presented in the sources
3. Capture Unique Insights: Note unique
    perspectives, case studies
4. Reference Completeness: Ensure all major
    topics are addressed

## Learning Objectives (Bloom's Taxonomy)
```

```
- Remember (LO1.x): Define, List, Identify,
    Name, Recall, State
- Understand (LO2.x): Explain, Describe,
    Summarize, Compare
- Apply (LO3.x): Calculate, Demonstrate,
    Solve, Apply, Execute
- Analyze (LO4.x): Analyze, Differentiate,
    Compare, Contrast
- Evaluate/Create (LO5.x): Evaluate, Design,
    Create, Judge

## PAGE COUNT RULES (based on Duration)
- 10-minute: 8-12 pages (6-9 lecture + 2-3
    quiz)
- 15-minute: 12-15 pages (9-12 lecture + 3
    quiz)
- 20-minute: 15-18 pages (12-15 lecture + 3-4
     quiz)
- 45-minute: 25-35 pages (20-28 lecture + 4-5
     quiz)
```

Content Outline Generation Prompt

```
The content outline breaks down the lesson
    plan into page-level
specifications. Each page has a clear
    instructional purpose.

## Page Types
- canvas: Lecture page for presenting
    teaching content
- quiz: Quiz page for assessing learning
    outcomes

## Design Rules
- Clear function: Each page has clear
    instructional purpose
- Appropriate density: Include substantive
    content
- Logical flow: Pages form natural teaching
    progression
- Objective traceability: Each page MUST
    specify LOs addressed

## CRITICAL REQUIREMENTS
1. Page 1 MUST be "Learning Objectives"
    canvas page
2. AT LEAST 3 quiz pages distributed
    throughout
3. Place quiz after every 4-5 canvas pages
4. Each quiz must assess LOs covered in
    preceding pages

## Output Format
{"title": "Course Title", "pages": [
  {"page_number": 1, "page_type": "canvas",
   "title": "Today's Learning Objectives",
   "content": "# Learning Objectives\n- LO1
    ...",
   "target_objectives": ["LO1.1", "LO2.1"],
   "cognitive_level": "remember"},
  {"page_number": 5, "page_type": "quiz",
   "title": null, "content": "## Practice",
   "target_objectives": ["LO1.1", "LO3.1"],
   "cognitive_level": "apply",
   "assessment_focus": "Test understanding of
    ..."}]}
```

J.3 Generator Agent Prompts

The Generator agent uses two main prompts for creating slides and quizzes, with detailed specifications for content depth and visual design.

Canvas Generation Prompt

```
You are a professional educational content
    designer,
generating standardized PPTist format Canvas
    components.

## Content Depth Requirements (CRITICAL)
Canvas content must have substantial depth,
    not simple lists.

### Content Expansion Principles
1. From bullet points to explanations: Each
    bullet should
   be a complete sentence (8-20 words)
2. Specificity principle: Include specific
    numbers, dates,
   names, places, and other details
3. Connectivity principle: Establish
    connections between
   concepts, helping students understand "why
    "
4. Application principle: Include at least 1
    real-world
   example or application scenario per page

### Content Depth Comparison
BAD:  "Algebra originated from Arabia"
GOOD: "Algebra comes from Arabic 'al-jabr',
    meaning
       'reunion of broken parts'"

## Visual Design Standards
Use standardized color schemes:
- Primary (titles): #1a5fb4 (dark blue)
- Secondary (subtitles): #3584e4 (medium blue
    )
- Accent (highlights): #ff7800 (orange)
- Text: #333333 (dark gray)

Typography Hierarchy:
- Main Title: 32-36px, Primary, Bold
- Subtitle: 24-28px, Secondary, Bold
- Body Points: 18-20px, Text, first word Bold
- Caption: 14-16px, #666666

## Canvas Constraints
Canvas size: 1000 x 562.5 (16:9 aspect ratio)
- Top margin: First element top >= 50
- Bottom safe zone: All elements bottom <=
    512.5
- Left/Right margin: >= 50

## Text Height Lookup Table (line-height=1.5)
| Font | 1 line | 2 lines | 3 lines | 4 lines
    |
| 18px |   49   |    76   |   103   |   130
    |
| 20px |   52   |    82   |   112   |   142
    |
| 24px |   58   |    94   |   130   |   166
    |
```

Quiz Generation Prompt

```
Quizzes are essential tools for assessing
    learning outcomes.

## Question Count Requirements (CRITICAL)

| Length    | Quizzes | Qs/Quiz | Total |
|-----------|---------|---------|-------|
| Short     |   2-3   |   5-6   | 15-18 |
| Medium    |   3-4   |   6-7   | 18-25 |
| Long      |    4    |   7-8   | 28-32 |

Per-Quiz Cognitive Level Distribution:
- 30% Remember level (definitions, facts,
    terminology)
- 40% Understand/Apply level (explanations,
    applications)
- 30% Analyze level (comparisons, synthesis)

## Question Design Principles
- Content Relevance: Based on teaching
    content only
- Appropriate Difficulty: Adjust for target
    audience
- Reasonable Options: Each distractor should
    be plausible
- Clear Explanations: Help students
    understand

## Question Types
1. single_choice: Only one correct answer
2. multiple_choice: Multiple correct answers

## Output Format
{"questions": [
  {"question": "What is the main site of
     photosynthesis?",
   "question_type": "single_choice",
   "options": [
    {"id": "A", "content": "Mitochondria"},
    {"id": "B", "content": "Chloroplast"},
    {"id": "C", "content": "Nucleus"},
    {"id": "D", "content": "Cell membrane"}],
   "correct_answers": ["B"],
   "explanation": "Chloroplasts contain
    chlorophyll to
    capture light energy for photosynthesis
    ..."}]}
```

Algorithm 1 TeachCraft Generation Pipeline

Require: Documents $\mathcal{D}$, Requirements $\mathcal{R}$
Ensure: Lesson $\mathcal{L} = (c_1, \ldots, c_m)$

// Explorer Agent $\mathcal{A}^E$ (Selection)
$\mathcal{C}_{\text{raw}} \leftarrow \textsc{Parse}(\mathcal{D})$ {Format-specific extraction}
for each chunk $c \in \mathcal{C}_{\text{raw}}$ **do**
  $\psi(c) \leftarrow \textsc{ClassifyIntent}(c, \mathcal{I})$ {6-intent taxonomy}
end for
$\mathcal{C} \leftarrow \{(c, \psi(c)) \mid c \in \mathcal{C}_{\text{raw}}\}$ {Intent-annotated pool}

// Planner Agent $\mathcal{A}^P$ (Sequencing)
$\mathcal{O} \leftarrow \textsc{ExtractLO}(\mathcal{C})$ {Learning objectives}

for each objective $o \in \mathcal{O}$ **do**
  $\beta(o) \leftarrow \textsc{AssignBloomLevel}(o)$ {Cognitive level}
end for
$\pi^* \leftarrow \textsc{Sort}(\mathcal{O}, \beta)$ {Bloom-ordered sequence}

$\mathit{plan} \leftarrow \textsc{GenerateLessonPlan}(\pi^*, \mathcal{R})$
$\mathit{outline} \leftarrow \textsc{GenerateOutline}(\mathit{plan}, \mathcal{C})$

// Generator Agent $\mathcal{A}^G$ (Synthesis)
$\mathcal{L} \leftarrow []$
for each page spec (t_j, o_j, β_j) in *outline* **do**
  $\mathcal{M}_j \leftarrow \{\mathit{plan}, \mathit{outline}, c_1, \ldots, c_{j-1}\}$ {Context memory}
  $\sigma_j \leftarrow (o_j, \beta_j, \mathit{terms}_j, \mathit{layout}_j)$ {Schema constraints}
  if $t_j =$ `Slide` **then**
    $c_j \leftarrow \textsc{GenerateCanvas}(\mathcal{M}_j, \sigma_j)$
  else
    $c_j \leftarrow \textsc{GenerateQuiz}(\mathcal{M}_j, \sigma_j)$
    assert $\exists\, s_i \prec c_j : o(s_i) = o(c_j)$ {Coherence check}
  end if
  $\mathcal{L}$.append(c_j)
end for
return $\mathcal{L}$