Reimagining Textbook Learning: An Interactive AI Tutor Approach Using Retrieval-Augmented Generation

Anonymous ACL submission

Abstract

We present **Textbook Tutor**, an interactive AI learning system that transforms static textbook chapters into structured learning experiences through a reproducible Textbook-to-Interaction pipeline. The system converts open educational resources into multiple instructional modes, including storytelling, business case simulations, and interactive challenges grounded in retrieval-augmented generation. Through controlled pipeline-level evaluations and a mixed-method user study (N=62), we find that instructional structure contributes more to short-term learning gains than increasing model capacity alone, and that lightweight lexical retrieval is sufficient for reliable chapter-level grounding. Guided by usability findings from an early prototype, we transitioned the system to a web-based deployment supporting both curated single-textbook tutoring and flexible multi-book workflows. Together, our results demonstrate that effective and accessible interactive textbook learning can be achieved without heavy infrastructure.

1 Introduction

Traditional textbook-based learning often struggles to support sustained engagement and conceptual understanding (Woo, 2009). Static chapters provide limited opportunities for interaction, feedback, or personalization, which can reduce motivation and slow learning progress, even for highly motivated students. These challenges are especially pronounced when learners must navigate dense material without adaptive guidance or structured practice. Recent advances in large language models have created new opportunities for educational support, including automated tutoring, summarization, and question answering (Woo, 2009). However, most AI-driven learning tools emphasize conversational question answering and rely on cloud-scale infrastructure, limiting both instructional diversity and accessibility in real-world educational settings. Prior systems such as Newtbot (Lieb and Goel, 2024) and Ruffle&Riley (Schmucker et al., 2023) demonstrate the potential of LLM-powered tutoring, but they often focus on a single interaction mode and require heavy infrastructure. As a result, there remains a need for modular, multi-mode tutoring systems that directly leverage open educational resources (OER) while remaining practical to deploy (OpenStax, 2023b; Lewis et al., 2021; Qin et al., 2023).

In this work, we study how retrieval-augmented generation (RAG) can be combined with instructional structure to transform static textbook content into interactive, pedagogically aligned learning experiences (Lewis et al., 2021). We hypothesize that, in low-resource educational settings, *instructional structure combined with lightweight retrieval grounding contributes more to learner usability and short-term learning outcomes than model scale alone*, building on prior work showing the importance of structured interaction in intelligent tutoring systems (Woo, 2009; Chaudhri et al., 2013) and recent evidence that retrieval-grounded tutoring can support effective learning without relying on large proprietary models (Slade et al., 2024). To test this design principle, we introduce an *Interactive AI Tutor* built around a reproducible *Textbook-to-Interaction (T2I)* pipeline that transforms chapter-structured textbook content into instructional formats, including storytelling modules, business case simulations, and interactive challenges grounded in retrieval-augmented generation (Lewis et al., 2021). Content creation is supported by a curated, chapter-aligned dataset generated using DeepSeek (AI, 2023a) and refined through manual validation to ensure fidelity to the source material.

An early research prototype enabled us to assess feasibility and learner experience under constrained computational settings. While this prototype demonstrated the promise of interactive, retrieval-grounded learning, it also surfaced practical limitations related to access time, technical setup, and usability for learners without programming experience. These insights motivated a human-centered redesign that transitions the tutor from a notebook-based research prototype to a deployed web-based system, with a focus on improving accessibility while preserving the core instructional pipeline. This work is guided by the following research questions:

1. How effectively can retrieval-augmented generation combined with DeepSeek-generated drafts transform static textbook content into coherent and pedagogically aligned instructional material?

2. How do multiple interactive learning modes, including storytelling, business case simulations, and structured challenges, influence learner engagement and short-term learning perceptions?

3. Is it feasible to deploy an AI-powered tutoring system in resource-constrained environments while maintaining usability and accessibility for non-technical users?

Contributions. This paper introduces (1) a reproducible Textbook-to-Interaction pipeline for transforming chapter-structured open educational resources into multiple instructional formats beyond dialogue-based tutoring; (2) a curated, chapter-aligned dataset generated using DeepSeek and refined through manual validation; (3) a human-centered transition from a notebook-based prototype to a deployed web-based learning system; and (4) an empirical evaluation combining pipeline analyses, a mixed-method user study, and deployment analysis.

2 Related Work

Intelligent tutoring systems (ITS) aim to provide adaptive and personalized learning experiences (Woo, 2009). Recent advances in large language models (LLMs) have enabled conversational tutoring, retrieval-grounded feedback, and dynamic content generation. However, many existing systems emphasize dialogue-centric interaction, assume cloud-based deployment, and provide limited support for chapter-aligned instruction over open educational resources (OER), leaving open questions around extensibility, reproducibility, and deployment feasibility.

LLM-powered tutoring systems. Conversational tutors such as **Newtbot** (Lieb and Goel, 2024), **Ruffle&Riley** (Schmucker et al., 2023), and **Khanmigo** (Khan Academy Labs, 2023) demonstrate the effectiveness of LLM-based dialogue for learning support, but primarily focus on conversational interaction and rely on cloud-scale or proprietary infrastructure. More recent work integrates retrieval-augmented generation (RAG) into tutoring systems: **Slade et al.** (Slade et al., 2024) and **Németh et al.** (Németh et al., 2024) report usability gains in coding and statistics education, yet provide limited analysis of how retrieval interacts with alternative instructional structures beyond dialogue or how such systems generalize to constrained deployment settings.

Interactive textbooks and retrieval-augmented models. Prior to LLMs, **Inquire Biology** (Chaudhri et al., 2013) illustrated the value of interactive textbooks grounded in curated knowledge bases, highlighting the importance of aligning instruction with source material. In parallel, retrieval-augmented generation has been shown to improve factual grounding in language models (Lewis et al., 2021). Broader retrieval-augmented and multimodal models, such as **RAMM** (Mialon et al., 2023), explore architectural advances but are not tailored to educational pipelines or chapter-level instructional design. Collectively, these works establish the promise of retrieval grounding, yet leave open how it can be combined with pedagogical structures such as storytelling, scenario-based reasoning, and structured challenges tied to OER.

Positioning. Prior work demonstrates the effectiveness of LLM-based tutors and retrieval grounding, but typically focuses on dialogue-only interaction and assumes substantial computational resources, limiting accessibility and extensibility. Few systems explore how multiple instructional modes interact with retrieval grounding within a single reproducible pipeline, or how such systems can be deployed in accessible, low-resource settings.

How our work extends this landscape. In contrast to prior LLM-based tutoring systems, our work examines how retrieval grounding can be systematically integrated with multiple instructional structures beyond dialogue within a single reproducible pipeline. We further analyze deployment under constrained resources, studying both curated

System / Paper	Modes	OER	Compute	Artifacts	Eval.	Notes
Newtbot (Lieb and Goel, 2024)	Dialogue	○	Heavy / Cloud	○	Usability	Secondary physics
Ruffle&Riley (Schmucker et al., 2023)	Dialogue	○	Heavy / Cloud	○	Mixed-methods	Design insights
Khanmigo (Khan Academy Labs, 2023)	Dialogue + tools	○	Heavy / Proprietary	○	Case studies	Large-scale platform
Slade et al. (Slade et al., 2024)	Dialogue + RAG	○	Mixed	Code	Usability	Coding courses
Németh et al. (Németh et al., 2024)	Dialogue + RAG	○	Mixed	○	Pilot	Statistics education
Inquire Biology (Chaudhri et al., 2013)	QA over KB	(non-LLM)	Lightweight	○	Classroom	Pre-LLM interactive textbook
RAMM (Mialon et al., 2023)	Multimodal gen.	○	Heavy / Research	Code	Benchmarks	Not education-specific
This work (T2I)	Story, Cases, Multiple Challenges + RAG (single/multi)	(OER + textbook PDFs)	Low local compute (CPU-based)	Dataset + Study Instruments + Code	User study + Retrieval	Web-based deployment

Table 1: Comparison of representative tutoring and RAG-in-education systems. "OER" indicates direct alignment with open educational resources.

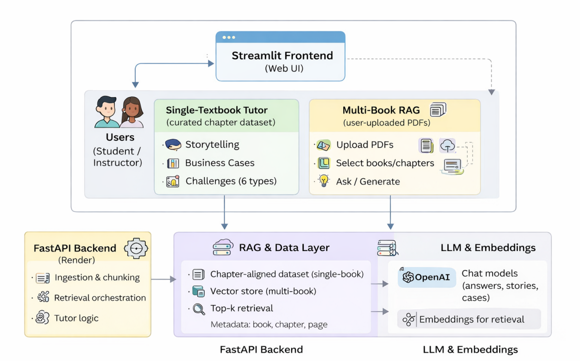

Figure 1: System overview of the Textbook Tutor platform. Textbook content is ingested and indexed through the Textbook to Interaction pipeline, enabling retrieval grounded generation of stories, business cases, challenges, and question answering. The architecture supports both a curated single textbook workflow and a flexible multi book workflow with user uploaded PDFs through a web based interface.

single-textbook tutoring and flexible multi-book workflows grounded in open educational resources.

3 Methodology

We design **Textbook Tutor** as a modular, retrieval-grounded learning system to study how static textbook chapters can be transformed into interactive, chapter-specific learning experiences under practical resource constraints. The system supports both a curated single textbook workflow and a flexible multi book workflow that ingests user uploaded PDFs. It is explicitly designed to operate under constrained computational settings while remaining accessible to non technical users through a web based interface.

Figure 1 provides an overview of the system architecture and data flow. The figure illustrates how textbook content is processed through the Textbook to Interaction pipeline, beginning with content ingestion and dataset preparation, followed by retrieval grounded generation, and delivered to learners through multiple interactive learning modes.

3.1 Data Preparation and Pipeline Overview

For the curated single textbook workflow, we extract chapters from the OpenStax *Workplace Software and Skills* textbook (OpenStax, 2023b) using PyMuPDF (Team, 2023). Chapters are segmented by section headers and cleaned to remove non instructional text. Using structured prompts with DeepSeek (AI, 2023a), we generate chapter summaries and up to five question answer pairs per chapter. All generated content is manually reviewed to ensure factual accuracy and alignment with the source material. The resulting chapter aligned dataset underpins storytelling, challenge generation, and guided question answering.

For the multi book workflow, users upload textbook PDFs through the web interface. Uploaded documents are processed through the same extraction and chunking pipeline, with metadata preserved at the book, chapter, and page level. This design enables consistent retrieval and generation behavior across both curated and user-provided textbooks, supporting direct comparison across workflows. The Textbook to Interaction pipeline consists of four stages. First, chapter aligned content is indexed using a lightweight retrieval layer designed for efficient similarity search under limited computational resources. In the curated workflow, retrieval relies on chapter anchored summaries and validated question answer pairs. In the multi book workflow, uploaded content is chunked and indexed using TF–IDF based representations (Salton and Buckley, 1988), enabling robust lexical retrieval without reliance on dense embedding infrastructure. Second, retrieved content is passed to a lightweight generation layer that produces grounded responses. Third, learners interact with the system through selectable learning modes. Fourth, learner activity is tracked at the session level to support progress visualization and analysis.

3.2 Interactive Learning Modes

The system supports multiple instructional modes designed to promote engagement and conceptual understanding. These include storytelling modules (sma, 2023) that transform chapter summaries into narrative explanations, business case (wri, 2023) simulations that map textbook concepts to realistic workplace scenarios, and six types of interactive challenges. The challenge types include flashcards, multiple choice questions, fill in the blank exercises, matching tasks, timed questions, and scenario based prompts with hints (Dicheva et al., 2015). All activities draw exclusively from validated chapter aligned content to reduce hallucination risk.

Chapter-aware question answering. In addition to predefined learning modes, the system provides an interactive question answering interface that allows learners to ask free-form questions about the selected textbook chapter. The assistant constructs a compact context from the chapter summary and validated question–answer pairs and generates responses grounded in this content. To support different learning preferences, learners can control the answer style, requesting concise summaries, step-by-step explanations, or real-world illustrative examples while keeping the chapter context fixed.

3.3 Deployment Architecture

The deployed system follows a client server architecture. The frontend is implemented using Streamlit and provides the learner interface for chapter selection, mode switching, and interaction. Backend services are implemented using FastAPI and manage document ingestion, retrieval, and generation requests. This separation enables scalable deployment while keeping the learner interface lightweight. For generation, the system supports models suitable for constrained environments, including quantized lightweight language models such as Falcon RW 1B (Institute, 2023), while larger models are used during development and validation. This design choice enables practical public deployment without dependence on specialized hardware. These design choices prioritize reliability and accessibility over model scale, aligning with our goal of evaluating interactive tutoring under realistic deployment constraints.

3.4 Prompting and Retrieval Strategy

We employ task specific prompt templates for summaries, storytelling, challenges, and question answering. At runtime, prompts are conditioned on retrieved chapter content and include explicit instructions to remain grounded in the provided text. Generation is constrained by token limits and conservative decoding settings to promote concise and faithful responses.

Retrieval is performed using lightweight similarity based search strategies over chapter aligned content. A fixed top k retrieval policy is applied without reranking. If no relevant content is retrieved above a similarity threshold, the system abstains from answering and instead prompts the learner to revisit relevant sections. This design prioritizes factual grounding and pedagogical reliability over answer coverage. This conservative retrieval and generation strategy is intentionally aligned with educational settings, where grounding and reliability are prioritized over open-ended response coverage.

4 Model Development

The model development of Textbook Tutor followed a staged design process. We first employed high-capacity language models during development to generate and validate instructional content. We then evaluated lightweight baseline models to test feasibility under constrained computational settings. Finally, the system was optimized for public deployment with an emphasis on accessibility and efficiency. This staged approach ensures that instructional quality is maintained without requiring heavy computation at runtime.

4.1 Development and Prototype Models

During early development, we employed large instruction-following models, including `mistralai/Mistral-7B-v0.1` (AI, 2023b) and `deepseek-ai/deepseek-coder-6.7b-instruct`, to generate chapter summaries, question–answer pairs, and draft instructional material. These experiments were conducted on a university high-performance computing cluster (64 CPU cores, 256 GB RAM, and up to 8 GPUs), enabling rapid iteration over prompt design and content validation.

The primary role of these models was offline content generation and dataset construction.All generated outputs were manually reviewed and refined before being incorporated into the instructional pipeline. These models are not used during learner-facing interaction. To evaluate feasibility in constrained environments and support early user studies, we implemented a notebook-based prototype in Google Colab using `tiiuae/falcon-rw-1b` (Institute, 2023). This model was selected for its relatively small size and stable performance on free-tier CPU and GPU resources. We applied 4-bit quantization using BitsAndBytes (Dettmers et al., 2022) to reduce memory usage and improve loading reliability. While Falcon-RW-1B provides weaker generative performance than larger models, it enabled interactive experimentation and empirical evaluation without specialized hardware, serving as a baseline for deployment feasibility.

4.2 Deployed System and Lightweight Generation

The final deployed system prioritizes accessibility, robustness, and scalability. Rather than relying on large language models at runtime, Textbook Tutor adopts a lightweight, retrieval-grounded generation strategy. For the curated single-textbook workflow, responses are constructed from chapter-aligned summaries and validated question–answer pairs. For the multi-book workflow, user-uploaded PDFs are indexed using TF-IDF-based retrieval, and responses are grounded in the most relevant retrieved chunks. This approach avoids dependence on dense vector indexes or continuous model inference, while preserving grounding and instructional fidelity.

4.3 Conversational Assistant

In addition to predefined learning modes, the system includes a chat-based, chapter-aware question answering assistant that allows learners to ask free-form questions about the selected textbook chapter. For each query, the assistant constructs a compact context from the chapter summary, validated question–answer pairs, and recent chat history, and generates responses using a hosted large language model via the OpenAI API. While the chapter content serves as the primary grounding signal (Lewis et al., 2021), the assistant may additionally use general world knowledge to provide intuitive real-world examples and analogies.

To accommodate different learning preferences, learners can control the answer style, requesting concise explanations, step-by-step reasoning, or real-world illustrative examples while keeping the chapter context fixed. The assistant is implemented using the `gpt-4o-mini` model and is intended to support exploratory learning. This component was not used in the user study, retrieval evaluations, or deployment analysis reported in this paper. We describe this component for completeness, but exclude it from all evaluations to avoid confounding instructional effects with proprietary models.

5 Evaluation

Evaluation Overview. We first present pipeline-level ablations isolating retrieval grounding, instructional structure, and model capacity, followed by a mixed-method user study validating usability and learner experience.

Note. An earlier ARR cycle included a 30-participant pilot; this revision reports the subsequent main study (N=62; 57 complete).

5.1 Pipeline-Level Ablation and Component Analysis

We conduct controlled pipeline-level ablations to isolate the effects of (i) prompt grounding, (ii) instructional structure, (iii) model capacity, and (iv) retrieval strength, holding all other components constant.

5.1.1 Prompt Grounding Ablation (RAG ON vs. OFF)

To isolate the effect of retrieval grounding in generation, we compare responses generated with retrieval enabled against generation-only baselines using identical prompts, models, and decoding parameters. Retrieval-augmented generation significantly improved rubric scores (mean total 3.9 vs. 2.7 out of 5; $\Delta = 1.2$, $p < 0.001$, $r = 0.62$), with substantial inter-rater agreement ($\kappa = 0.71$).

5.1.2 Instructional Structure Ablation (MCQ vs. Multi-Mode)

To isolate the effect of instructional structure independent of content and model configuration, we compare single-mode interaction (MCQ-only) against multi-mode interaction (MCQ, storytelling, and scenario-based challenges). Participants exposed to multi-mode learning showed larger objective concept-quiz gains than MCQ-only interaction (median +2/8 vs. +0/8; $p = 0.004$, $r = 0.49$).

5.1.3 Model Capacity Ablation

To assess the impact of model scale, we compare a lightweight model (Falcon-RW-1B) with a larger model (Mistral-7B) under identical prompts, retrieval inputs, and decoding settings. Larger models outperformed lightweight baselines on faithfulness (exact-overlap 68% vs. 51%) and overall rubric scores (+0.6/5; $p = 0.02$).

Table 2: Chapter-level retrieval performance on the merged chapter dataset (486 queries).

Retrieval Method	Recall			Precision		
Precision / Recall	@1	@3	@5	@1	@3	@5
TF–IDF	0.947	0.986	0.992	0.947	0.947	0.945
BM25	**0.977**	**0.992**	**0.992**	**0.977**	**0.979**	**0.980**
Hybrid (TF–IDF + BM25)	0.965	0.988	0.992	0.965	0.964	0.963

5.1.4 Retrieval Strength Ablation

To evaluate whether lightweight retrieval is sufficient for grounding, we compare TF–IDF, BM25, and hybrid lexical retrieval strategies while holding generation fixed. Results show consistently high chapter-level recall and precision across all methods, indicating that lightweight lexical retrieval is sufficient for reliable chapter-level grounding.

Single-Book We evaluate the retrieval component independently using a chapter-aligned dataset derived from the curated single-textbook workflow. The evaluation set consists of **486 queries** generated from validated chapter-level questions. Each query is associated with a gold chapter, and retrieval is considered successful if at least one retrieved chunk belongs to the gold chapter.

We compare three lightweight retrieval strategies designed for low-resource settings: TF–IDF, BM25, and a hybrid lexical retriever combining both. Results are shown in Table 2.

Multi-Book Routing We evaluate multi-book routing using two complementary textbooks: *Beyond Coding* (CAUL, 2023) and *Computer Science* ((OpenStax, 2023a)). We construct manually curated query sets of **100 queries** and **250 queries**, respectively, each labeled with its gold source textbook. Retrieval is considered correct if at least one of the top-k retrieved chunks originates from the correct textbook.

We report **Recall@k** for $k \in \{1, 3, 5\}$ and compare three lightweight retrieval strategies suitable for low-resource deployment: TF–IDF, BM25, and a hybrid lexical retriever. Results are shown in Table 3.

Across both datasets, Recall@5 exceeds **0.80** in all configurations and approaches **1.0** for the more structured *Computer* textbook. TF–IDF performs best at rank 1 for semantically phrased queries, BM25 provides stronger coverage at higher ranks, and the hybrid strategy consistently balances these strengths, achieving the best or near-best Recall@1.

Table 3: Multi-book retrieval performance (Recall@k) across two textbooks.

Method	Recall@1	Recall@3	Recall@5
Beyond Coding (CAUL, 2023)(100 queries)			
TF–IDF	0.294	0.686	0.765
BM25	0.098	0.765	0.843
Hybrid	0.235	0.725	0.804
Computer (OpenStax, 2023a) (250 queries)			
TF–IDF	0.504	0.972	0.984
BM25	0.504	**0.984**	**0.996**
Hybrid	**0.516**	0.976	0.992

5.2 User Study: Human-Centered Validation (Colab Prototype)

5.2.1 Study Design

We conducted a mixed-method evaluation to assess the tutor's usability, engagement, and instructional effectiveness. The study protocol was approved by the university IRB (**IRB2025-0182**), and all participants provided informed consent. A total of **62 participants** accessed the system, and **57 completed** both pre- and post-assessments. Participants were students or early-career professionals in computing-related fields, recruited through university mailing lists, social media (LinkedIn, WhatsApp), and faculty announcements. Participation was voluntary and unpaid.

Each participant completed a single **45-minute** session in **Google Colab**. A pilot study with **5 participants** helped refine instructions and survey items. Sessions followed a consistent flow: (1) a **pre-assessment** on AI familiarity and comfort, (2) interaction with the tutor across storytelling, business case, and challenge modes, and (3) a **post-assessment** on perceived learning, usability, and experience. Participants completed fixed pre- and post-assessment instruments (full questionnaires in Appx. F and Appx. G); both instruments were IRB-approved and administered immediately before and after the session.

To ensure a consistent experience, participants received both a full guide[1] and a concise mini guide[2], and embedded prompts within the notebook offered step-by-step guidance during runtime. Data collected included Likert-scale ratings, open-ended feedback, and system logs (session duration, number of activities completed, and interaction frequency). Pre/post items targeted two short-term learning proxies—*familiarity* and *com-*

[1]Anonymized documentation link provided in supplementary material.

[2]Anonymized documentation links provided in supplementary material.

Table 4: Post-assessment descriptive statistics (N = 57). Likert-scale items rated 1–5.

Feedback Item	Mean	Median	SD
Overall experience	4.70	5.0	0.82
Ease of using the interface	4.60	5.0	0.94
Helpfulness of walkthroughs	4.70	5.0	0.82
Engagement of learning modes	4.70	5.0	0.50
Content clarity and quality	4.76	5.0	0.47
Ease of session completion	4.78	5.0	0.46
Would use again (1–5)	4.71	5.0	0.50

Table 5: Pre- vs. post-assessment comparisons (Wilcoxon signed-rank tests).

Comparison	Pre	Post	Statistic	p-value
Familiarity to Understanding	2.39	2.77	88.0	0.00003
Comfort to Experience	2.38	2.82	102.0	0.00061

fort—alongside post-only usability and engagement items. We treat these measures as self-report proxies rather than objective learning gains. Quantitative analyses employed descriptive statistics, exploratory factor analysis, and non-parametric tests (Wilcoxon signed-rank, Mann–Whitney U); ordinal and nominal items are reported using frequencies rather than means. The conversational question answering assistant was not supported in the Colab prototype and was disabled during the study to ensure a consistent and controlled interaction flow across participants.

5.2.2 Usability and Engagement

Post-assessment feedback (Table 4) indicated strong positive reception. Mean ratings across all Likert-scale items exceeded **4.5**, with participants reporting high ease of use (**4.60**), content clarity (**4.76**), and satisfaction with guided walkthroughs (**4.70**). Engagement of learning modes (**4.70**) and overall experience (**4.70**) were similarly rated. A total of **55 out of 57 participants (96%)** reported improved understanding, and willingness to reuse the tutor scored **4.71**, indicating strong perceived value.

Exploratory factor analysis revealed two latent factors: *Usability* and *Engagement and Content Quality*. The Kaiser–Meyer–Olkin value was **0.674**, and Bartlett's test was significant ($\chi^2 = 114.66$, $p < 0.001$), confirming data suitability. Together, the factors explained **50.27%** of the variance. Internal consistency was excellent for Usability ($\alpha = 0.895$), while Engagement and Content Quality showed modest reliability ($\alpha = 0.449$). Given the heterogeneous nature of this construct and the small item set, we treat this factor structure as exploratory and report full loadings in Appendix E.

5.2.3 Learning Outcomes

Wilcoxon signed-rank tests demonstrated significant improvements between pre- and post-assessments (Table 5). Self-reported familiarity increased from **2.39** to **2.77** ($p < 0.001$), while comfort increased from **2.38** to **2.82** ($p < 0.001$), indicating improved conceptual understanding and confidence.

No significant differences were observed across user subgroups based on initial familiarity or comfort (all $p > 0.42$), and effect sizes were negligible ($|\delta| < 0.15$), suggesting consistent improvements regardless of prior experience.

5.3 Deployment Analysis

While the Colab prototype enabled controlled evaluation under constrained resources, it introduced practical barriers including session time limits, dependency setup, and the need for notebook interaction. Several participants reported startup delays and difficulty navigating runtime cells, consistent with observed session logs.

Motivated by these findings, we redesigned and deployed the system as a web-based platform using a Streamlit frontend and a FastAPI backend. The deployed system eliminates installation and notebook execution requirements, supports one-click browser access, and enables both a curated single-textbook workflow and a flexible multi-book workflow with user-uploaded PDFs. To ensure accessibility in low-resource environments, the deployed system operates entirely on CPU using lightweight TF–IDF retrieval and template-driven generation, avoiding reliance on dense vector indexes or continuous large-model inference. Although we did not conduct a second user study, the deployment directly addresses the dominant usability limitations observed in the baseline evaluation and enables browser-based conversational question answering.

6 Discussion

This section interprets the findings in relation to the research questions and the design of the Textbook-to-Interaction pipeline, emphasizing the relative contributions of instructional structure, retrieval grounding, and deployment constraints as evidenced in Section 5.

RQ1: Effectiveness of retrieval grounded content generation. Results from the objective pipeline evaluations show that retrieval augmented generation substantially improves factuality and grounding compared to generation without retrieval (Table 9). In addition, chapter level retrieval experiments demonstrate consistently high recall and precision across both curated and multi book settings (Table 2). These findings indicate that lightweight lexical retrieval, when combined with chapter aligned summaries and validated question answer pairs, is sufficient to support reliable and pedagogically aligned instructional generation.

RQ2: Impact of multiple interactive learning modes. Evidence from the baseline user study indicates that learners respond positively to exposure across multiple instructional formats. Post assessment results show high engagement and usability ratings across storytelling, business case simulations, and structured challenges (Table 4). Moreover, the objective concept quiz comparison demonstrates that multi mode interaction yields larger learning gains than multiple choice questions alone (Table 10). Together, these results suggest that varied instructional structures support deeper conceptual engagement even within short learning sessions.

RQ3: Feasibility of low resource deployment. Evaluation results and participant feedback highlight that while the notebook based prototype enabled controlled study, it introduced usability barriers related to setup and session management. The redesigned web based deployment directly addresses these limitations by removing installation requirements and operating entirely on CPU with lightweight retrieval and template driven generation. Although no second user study was conducted, the deployment aligns closely with the usability issues identified in Section 5 and demonstrates practical feasibility for non technical users.

Implications. Overall, the evaluation shows that effective interactive tutoring can be achieved without heavy infrastructure when retrieval grounding, instructional structure, and deployment constraints are jointly considered. The Textbook to Interaction pipeline provides a reusable framework for transforming open educational resources into accessible and engaging interactive learning experiences.

7 Limitations and Future Work

Despite strong usability and engagement results, this work has several limitations. First, the controlled user study was conducted on a notebook-based prototype; while its findings directly informed the redesigned web-based system, we did not conduct a second quantitative user study after deployment. Second, learning outcomes were measured primarily using short-term self-report proxies, supplemented by a small objective concept quiz, and do not capture long-term learning or retention. Third, the system prioritizes grounding and reliability through constrained retrieval and generation, which may limit coverage for highly open-ended or cross-document queries. Finally, this work does not include direct comparisons with other tutoring systems or commercial AI assistants. We did not include a between-system baseline due to the difficulty of deploying comparable tutoring systems under identical low-resource constraints.

Future work will include longitudinal and classroom-based evaluations, adaptive retrieval strategies that better balance flexibility and grounding, comparative studies with alternative tutoring approaches, improvements to the efficiency and responsiveness of the conversational assistant, and extensions to the web platform such as persistent user accounts, instructor analytics, and support for additional subjects and multimodal content.

8 Conclusion

This paper presented an Interactive AI Tutor that transforms static textbook content into structured, interactive learning experiences using a reproducible Textbook-to-Interaction pipeline. The system combines lightweight retrieval grounding with multiple instructional modes, including storytelling, case-based learning, structured challenges, and question answering. Evaluation of an early prototype showed high usability, strong engagement, and improvements in short-term learning proxies, motivating a human-centered transition to a web-based deployment. Overall, this work demonstrates that effective and accessible interactive tutoring can be achieved without heavy infrastructure, and provides a practical framework for enhancing open educational resources. Future work will explore longer-term learning outcomes, real-world classroom integration, and broader applicability across domains and modalities.

Ethics Statement

The user study was conducted with voluntary participation under university-approved IRB protocol. No personally identifiable information was collected.

Broader Impact

This work aims to improve access to interactive learning using open educational resources and lightweight AI deployments. It can benefit learners in resource-constrained environments by enhancing personalization and engagement. Limitations include dependency on internet access and evolving AI model reliability.

References

2009. Copyright page. In Beverly Park Woolf, editor, *Building Intelligent Interactive Tutors*, page iii. Morgan Kaufmann, San Francisco.

2023. How to write a business case. Accessed: 2025-04-28.

2023. Sell your product with powerful storytelling. Accessed: 2025-04-28.

DeepSeek AI. 2023a. Deepseek coder: Code large language model. https://huggingface.co/deepseek-ai/deepseek-coder-6.7b-instruct. Accessed: 2025-04-28.

Mistral AI. 2023b. Introducing mistral 7b. https://mistral.ai/news/introducing-mistral-7b/. Accessed: April 26, 2025.

CAUL. 2023. *Beyond Coding: Professional and Soft Skills for Computing*. OER Collective.

Vinay K. Chaudhri, David Gunning, Hannah Lane, Peter Clark, Ken Barker, Simon Chaw, Mark Greaves, Benjamin Grosof, Alex Leung, Suchi Mishra, Julio Pacheco, Bruce Porter, Aaron Spaulding, Dan Tecuci, and Peter Z. Yeh. 2013. Inquire biology: A textbook that answers questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 27, pages 1624–1631. AAAI Press.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.

Darina Dicheva, Christo Dichev, Gennady Agre, and Galia Angelova. 2015. Gamification in education: A systematic mapping study. *Educational Technology & Society*, 18(3):75–88.

Technology Innovation Institute. 2023. Falcon-rw-1b model card. https://huggingface.co/tiiuae/falcon-rw-1b. Accessed: 2025-04-28.

Khan Academy Labs. 2023. Introducing Khanmigo: AI-Powered Tutor and Teaching Assistant. https://www.khanacademy.org/khan-labs. Accessed: 2025-04-25.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Anna Lieb and Toshali Goel. 2024. Student interaction with newtbot: An llm-as-tutor chatbot for secondary physics education. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24, New York, NY, USA. Association for Computing Machinery.

Grégoire Mialon, Roberto Dessì, Maria Lomeli, Christoforos Nalmpantis, Ram Pasunuru, Roberta Raileanu, Baptiste Rozière, Timo Schick, Jane Dwivedi-Yu, Asli Celikyilmaz, Edouard Grave, Yann LeCun, and Thomas Scialom. 2023. Augmented language models: a survey. *Preprint*, arXiv:2302.07842.

Renáta Németh, Annamária Tátrai, Miklós Szabó, and Árpád Tamási. 2024. Using a rag-enhanced large language model in a virtual teaching assistant role: Experiences from a pilot project in statistics education. *Hungarian Statistical Review*, 7(2):3–27.

OpenStax. 2023a. *Computer Science*. Rice University.

OpenStax. 2023b. *Workplace Software and Skills*. OpenStax, Rice University.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *Preprint*, arXiv:2302.06476.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523.

Robin Schmucker, Meng Xia, Amos Azaria, and Tom Mitchell. 2023. Ruffle&riley: Towards the automated induction of conversational tutoring systems. In *NeurIPS 2023 Workshop on Generative AI for Education*.

Joseph J. Slade, Alina Hyk, and Regan A. R. Gurung. 2024. Transforming learning: Assessing the efficacy of a retrieval-augmented generation system as a tutor for introductory psychology. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 68, pages 1827–1830, Los Angeles, CA, USA. SAGE Publications.

MuPDF Team. 2023. Pymupdf: Lightweight pdf parsing and rendering. https://github.com/pymupdf/PyMuPDF. Accessed: 2025-04-26.

A Appendix

B Core Textbook-to-Interaction Pipeline

Textbook PDF T Interactive AI Tutor Experience

Step 1: Chapter Extraction

Extract chapter-wise content $C = \{c_1, c_2, ..., c_n\}$ from T using PDF parsing tools.

Step 2: QA Dataset Generation

chapter $c_i \in C$ Generate question–answer pairs QA_i and summary S_i using DeepSeek Coder-6.7B Store initial dataset $D = \{(QA_i, S_i, c_i)\}$

Step 3: Manual Review and Cleaning

Filter for hallucinations, formatting issues, and factual errors to produce cleaned dataset D^*

Step 4: Semantic Indexing

Embed QA_i with MiniLM and store in FAISS vector index V

Step 5: Retrieval-Augmented Generation

Use LangChain with index V to enable context-aware content retrieval and generation

Step 6: Model Integration

Internal high-resource environment Load Mistral-7B Low-resource environment (e.g., Colab) Load Falcon-RW-1B with 4-bit quantization using BitsAndBytes

Step 7: User Interaction Layer

Initialize user profile and preferences

If user is new or walkthrough requested **then**

Launch guided walkthrough with:

- Step-by-step preview of all challenge types
- Navigation controls (next/back/skip)
- Visual prompts and tooltips

Else

Launch interactive learning modules:

- Storytelling modules
- Business case simulations
- Interactive challenges:
  - Flashcards
  - Multiple-choice quizzes (MCQs)
  - Fill-in-the-blank tasks
  - Matching exercises
  - Timed questions
  - Scenario-based challenges with hints

Step 8: Gamification and Feedback

Track XP, level progression, and badge collection Provide real-time feedback and update learner dashboard

C Operational metrics

D Baseline and Ablation Details

D.1 Rubric (0–5 scale)

We evaluated generated responses on three criteria: (1) **Factuality** (0–2), measuring correctness relative to the textbook source; (2) **Grounding/Traceability** (0–2), measuring explicit linkage to retrieved spans; and (3) **Clarity** (0–1), assessing readability and coherence. Total possible score: 5. Table 8 details the rubric.

D.2 Rater Agreement

Two independent raters scored all outputs using the rubric above. Inter-rater reliability was $\kappa = 0.71$ (95% CI [0.62, 0.80]), indicating substantial agreement. Disagreements were resolved through adjudication.

D.3 RAG ON vs. OFF Results

D.4 Single vs. Multi-Mode Results

D.5 Model Comparison Results

D.6 Concept Quiz Items (Objective)

Below are representative items from the 8-question concept quiz used in the single- vs. multi-mode study.

- Q1: What is retrieval-augmented generation (RAG) and how does it improve factual grounding?
- Q2: Describe two benefits of 4-bit quantization in low-resource deployments.
- Q3: In the storytelling mode, what instructional purpose do reflective questions serve?
- Q4: What role does the FAISS index play in semantic retrieval?

E Exploratory Factor Analysis Details

E.1 Factor Loadings and Communalities

Table 12 reports the factor loadings for post-assessment items based on principal axis factoring with varimax rotation. Two factors emerged, explaining a total of **50.27%** of the variance (**37.65%** for Factor 1 and **12.62%** for Factor 2). Factor 1 corresponds primarily to *Usability*, while Factor 2 captures *Engagement and Content Quality*.

Template	Intent / Inputs	Full Prompt Text (Simplified)
QA Generation	*Intent:* Create up to N factual QA pairs from a chapter excerpt. *Inputs:* {chapter_title}, {chapter_excerpt}, {N} (default 5)	**System:** You are an educational content author. Produce concise, factual question–answer pairs strictly grounded in the provided chapter excerpt. Avoid speculation. **User:** Chapter: {chapter_title}. Excerpt: {chapter_excerpt}. Task: Generate up to {N} QA pairs. Each pair must be answerable solely from the excerpt. Use JSON schema: `[ { "question": "...", "answer": "...", "evidence_span": "..." }, ... ]`. If information is insufficient, omit that pair.
Summary Generation	*Intent:* Produce a short chapter summary. *Inputs:* {chapter_title}, {chapter_excerpt}, {max_words} (default 150)	**System:** You are summarizing textbook material for learners. Stay faithful to the source; no external facts. **User:** Chapter: {chapter_title}. Excerpt: {chapter_excerpt}. Task: Write a concise summary ({max_words} words) capturing key definitions, processes, and takeaways. Include no references beyond the excerpt.
Storytelling Scaffold	*Intent:* Transform a validated summary into a short workplace scenario. *Inputs:* {chapter_title}, {validated_summary}, {persona_A}, {persona_B}	**System:** You are an instructional designer. Create a realistic, workplace-relevant micro-story aligned with the provided summary. **User:** Summary: {validated_summary}. Task: Write a 120–180 word narrative featuring {persona_A} and {persona_B}. The scenario must exemplify 2–3 core ideas from the summary (**bold key terms on first mention**). End with a reflective question answerable from the narrative. Output JSON: `{"story": "...", "reflective_question": "..." }`.

Table 6: Prompt templates used in the Textbook-to-Interaction (T2I) pipeline. Braced fields are filled at runtime.

Setting / Rule	Public (Falcon-RW-1B)	Dev (Mistral-7B)	Notes
Decoding temperature	0.7	0.3–0.5	Higher for storytelling; lower for QA
Top-p (nucleus)	0.9	0.9	Paired with moderate temperature
Top-k	50	50	Conservative cap to reduce drift
Max new tokens	256	256 (QA), 180 (story)	Enforced at pipeline level
Repetition penalty	1.10	1.10	Mitigates loops
Stop sequences	`{"\n\n"}`	same	Prevent runaway JSON / paragraphs
Grounding mode	Required	Required	Answers must cite retrieved span IDs
Similarity threshold θ	0.35 cosine	0.35 cosine	If $\max$ sim $< \theta$, abstain
Abstention response	"I don't have enough context to answer that."	same	Followed by one clarifying question
Evidence requirement	At least one quoted / cited span per key claim	same	Regex check before output
External web access	Disabled	Disabled	Prevent leakage beyond OER
Story length guard	120–180 words	120–180 words	Keeps mode parity
JSON schema checks	Strict (pydantic)	Strict	QA / story outputs validated

Table 7: Guardrails and decoding defaults. Values shown are the defaults used in our experiments unless otherwise noted.

Criterion	Scoring Description
Factuality (0–2)	0 = Incorrect, 1 = Partially correct, 2 = Fully correct
Grounding (0–2)	0 = No evidence cited, 1 = Partial citation, 2 = Explicit and correct citation
Clarity (0–1)	0 = Unclear or incoherent, 1 = Clear and coherent

Table 8: Rubric used to score generated outputs in baseline evaluations.

Metric	RAG ON	RAG OFF
Mean total score (0–5)	3.9	2.7
Factuality	1.8	1.2
Grounding	1.7	0.8
Clarity	0.4	0.7

Table 9: Comparison of assistant performance with and without retrieval grounding (N=40 questions).

E.2 Communalities and Reliability

Given the heterogeneous nature of the "Engagement/Content Quality" construct and the small item set, internal consistency was modest ($\alpha = 0.45$). We therefore treat the two-factor solution as exploratory. Future work will refine the item pool and validate the structure on a larger sample.

F Pre-Assessment Questionnaire

Participants completed the following pre-assessment form prior to using the AI Tutor. The form collected demographic information, prior experience with textbook learning and AI tools, learning preferences, and expectations from the system.

F.1 Section 1: Basic Information

- What best describes your current role?
  - High school or less
  - Associate's degree
  - Bachelor's degree
  - Master's degree
  - PhD degree
  - Instructor/Professor
  - Working Professional
  - Other

Measure	MCQ Only	MCQ+ Story+ Scenario
Median concept quiz delta (0–8)	+0	+2
p-value (Wilcoxon)	0.004	
Effect size r	0.49	

Table 10: Learning gains with single- vs. multi-mode interaction (n = 20).

Metric	Falcon-RW-1B	Mistral-7B
Exact overlap with retrieved spans	51%	68%
Mean rubric total (0–5)	3.2	3.8
p-value	0.02	

Table 11: Model comparison on 30 held-out prompts.

Item	Factor1 (Usability)	Factor2 (Engagement /Content)
Overall experience	0.885	0.026
Ease of using the interface	0.881	0.224
Helpfulness of walkthroughs	0.833	-0.160
Engagement of learning modes	-0.011	0.599
Content clarity and quality	0.076	0.512
Ease of session completion	-0.009	0.246

Table 12: Exploratory factor loadings for post-assessment items ($N = 57$). Principal axis factoring with varimax rotation. Total variance explained: 50.27%.

- How familiar are you with using AI-based educational tools? (Very familiar / Somewhat familiar / Not familiar at all / Other)

F.2 Section 2: Previous Experience with Textbook Learning

- How often do you use textbooks for learning? (Daily / Weekly / Occasionally / Rarely / Never)
- Have you used any interactive or digital learning platforms before? (Yes / No)

F.3 Section 3: Experience with Digital Learning Platforms

- Please list any digital learning platforms you've used.

F.4 Section 4: Learning Preferences & Expectations

- Which learning methods do you prefer? (Reading text / Watching videos / Interactive quizzes / Case-based learning / Storytelling / Group discussions)
- What are your main learning goals with the AI Tutor?
- What do you expect from an AI-powered educational tool?

F.5 Section 5: Technical Readiness

- How comfortable are you with using tools like Google Colab or Jupyter Notebooks? (Very comfortable / Somewhat comfortable / Not comfortable)
- Have you used any AI-driven tools before? (Yes / No / Maybe)

F.6 Section 6: Experience with AI Tools

- If yes, please describe the tools you've interacted with and your experience using them.

F.7 Section 7: Expectations from the AI Tutor

- What kind of content do you expect the AI Tutor to help you with? (Subject-specific knowledge / Study skills / Test preparation / Real-world applications / Problem-solving)
- Do you prefer a personalized or standardized learning experience? (Personalized / Standardized / No preference)
- What features would you want in an AI-powered learning tool?

G Post-Assessment Questionnaire

After completing their interaction with the AI Tutor, participants filled out the following post-assessment questionnaire. The form was designed to evaluate usability, user engagement, content quality, technical performance, and overall experience. Responses informed our analysis of short-term learning impact and user satisfaction.

G.1 Section 1: Basic Information

- What best describes your current role?
  - High school or less
  - Associate's degree
  - Bachelor's degree
  - Master's degree
  - PhD degree
  - Instructor / Professor
  - Working Professional
  - Other
- How did you access the AI Tutor?
  - Through Google Colab directly
  - Researcher sent me the link
  - Other

G.2 Section 2: Overall Experience

- How would you rate your overall experience with the AI Tutor?
  (1 = Very Poor . . . 5 = Excellent)
- How easy was it to use the AI Tutor interface?
  (1 = Very Difficult . . . 5 = Very Easy)
- How helpful were the guided walkthroughs in getting started?
  (1 = Not helpful . . . 5 = Very helpful)
- How engaging did you find the different learning modes (Business Case, Storytelling, Challenges)?
  (1 = Not Engaging . . . 5 = Very Engaging)

G.3 Section 3: Challenge Types

- Which challenge types did you try?
  - Flashcards
  - Multiple-Choice Questions (MCQ)
  - Fill in the Blanks
  - Match the Answers
  - Timed Questions
  - Scenario-Based Challenges
- Which challenge type was the most effective for your learning? Why?
- Was the feedback (XP points, correct/incorrect indicators, progress tracking) helpful in understanding your performance? (Yes / No / Maybe)

G.4 Section 4: Performance and Content

- How would you rate the quality and clarity of the content presented in the AI Tutor?
  (1 = Very Poor . . . 5 = Excellent)
- Overall, how easy was it to complete your learning session with the AI Tutor?
  (1 = Very Difficult . . . 5 = Very Easy)
- Did the AI Tutor help improve your understanding of the concepts?
  - Yes, significantly
  - Yes, somewhat
  - Neutral
  - No, not really
  - No
- How would you describe the speed and performance of the AI Tutor in Google Colab?

- Fast and smooth
- Acceptable
- A bit slow
- Very slow or laggy
- Didn't work properly

G.5 Section 5: Suggestions and Future Use

- What features would you like to see added in the future?
- Would you use the AI Tutor again if it were available as a public website? (Yes / No / Maybe)
- Any other comments or suggestions to improve the AI Tutor?

H XP Schedules and Level Thresholds

H.1 XP Reward Schedule

Table 13 shows the XP reward structure used in the prototype implementation. Points are awarded for completing different interactive challenge types, and their share of total XP reflects relative contribution to learner progression.

Challenge Type	XP per Attempt	Share of Total XP (%)
Flashcards (Flip)	5	7.5
MCQ quiz	10	14.9
Fill-in-the-Blank	10	14.9
Match-the-Answers	12	17.9
Timed questions	15	22.4
Scenario-based (Hint)	15	22.4

Table 13: XP reward schedule for interactive challenges.

H.2 Level Progression Thresholds

Learners advance through levels as they accumulate XP. Table 14 lists the XP requirements for each level and the cumulative total needed for progression.

Level	XP to Next	Cumulative XP
$1 \rightarrow 2$	50	50
$2 \rightarrow 3$	75	125
$3 \rightarrow 4$	100	225
$4 \rightarrow 5$	125	350
$5 \rightarrow 6$	150	500
$6 \rightarrow 7$	175	675
$7 \rightarrow 8$	200	875
$8 \rightarrow 9$	225	1100
$9 \rightarrow 10$	250	1350
$10 \rightarrow 11$	275	1625

Table 14: XP thresholds for each level in the gamification engine.