EduMap: An Interactive Mind-Map Generator for Long Academic PDFs with Adaptive Highlighting

Yana Jin, Nicole Vu, Farhana Anjum, Shaun Ting
P(What's Next) = 1

Abstract

Traditional PDF viewers present content in a strictly linear format, forcing readers to mentally reconstruct complex document hierarchies. This cognitive load often hinders the identification of key concepts and their relationships. We present EduMap, an interactive system that combines layout and multimodal-aware PDF extraction with large language model (LLM) orchestration to produce hierarchical markdown and an editable mind map.

The interface couples a markdown editor with a React Flow canvas so structure stays synchronized. Nodes can carry LLM-assigned priority and modality tags, and cross-pane highlighting links selections between the outline and the graph. Users can export both the current edited map and a frozen original AI baseline (markdown and PNG snapshot). Our evaluation demonstrates that EduMap compares favorably to baselines on conceptual accuracy, hierarchical organization, and study usefulness.

1 Introduction

1.1 Problem Definition

The core problem we address is how to automatically transform long, dense academic PDFs into an interactive, structured representation that supports comprehension and prioritization. Currently, academic documents lack visual structure, forcing readers to process content linearly without seeing how topics relate or nest within each other. Our objective is to extract the hierarchical organization of a document, identify key concepts, and estimate the relative importance of different sections to present them as an editable mind map.

1.2 Background and Motivation

Students and researchers frequently read long academic PDFs that contain dense, hierarchically structured information. However, traditional viewers require readers to mentally reconstruct this structure while simultaneously processing complex ideas. It is currently difficult to recognize areas that require deeper cognitive effort, such as formal definitions or intricate arguments. This results in significant time spent navigating and re-reading without forming a clear mental model.

1.3 Current Practice and Limitations

Existing tools like NotebookLM and various academic summarizers generate condensed text or provide question-answering capabilities. However, they produce static outputs and do not preserve the structural relationships within a document in a visually navigable form. Furthermore, most tools treat all content uniformly, failing to distinguish between foundational concepts and highly technical sections. The primary differentiator of EduMap is its ability to preserve document hierarchy within an editable graph format, rather than providing a simple text-based summary.

1.4 Target Audience and Impact

This work is primarily for students, educators, and researchers who manage high volumes of technical literature. By shifting from passive reading to interactive knowledge mapping, EduMap supports more efficient learning, allows for personalized study artifacts, and helps users prioritize cognitive effort where it is most needed.

2 Literature Survey

We have integrated the latest research findings from the fields of document intelligence and text summarization to inform our approach. First, Ke et al. (2026) provide a systematic survey of Large Language Models (LLMs) in document intelligence, tracing the technological evolution from traditional OCR and layout analysis to multimodal models and generative LLMs. This macro-perspective provides a comprehensive technical framework for processing complex documents, assisting us in building

an end-to-end pipeline from long-text parsing to structured output.

For the specific structural analysis phase, we utilize the LiLTv2 model proposed by Wang et al. (2025). By jointly modeling textual content and spatial layout information, this model enables the precise identification of hierarchical structures and key fields within visually rich documents, providing core technical support for extracting data from complex layouts such as academic papers and courseware.

Regarding information filtering and extraction of key content, we refer to the systematic review of extractive text summarization by Azam et al. (2025). Using their standardized scoring and ranking mechanisms, we can efficiently locate core concepts and key sentences from the parsed text, establishing the content foundation for mind map construction.

Finally, in the stage of converting extraction results into structured Markdown format, we incorporate the research by Colakoglu et al. (2026) on the design space for LLMs handling layout-rich documents. By optimizing input representations, chunking strategies, and prompt designs, we enable the model to more accurately understand document hierarchies and integrate text with layout information into a well-formatted and logically clear structured output.

3 Approach / Methods

We have developed a web application along with supporting structural processing of natural language and also multimodal extraction from the uploaded PDFs. Our application EduMap provides an AI-driven extraction pipeline that processes the PDFs to extract the atomic concepts and organize them into a logical hierarchy.

3.1 Hypothesis and Strategy

The design of EduMap rests on a two-stage extraction process: first gather concepts, then organize them into a hierarchy. We hypothesized that separating these steps would produce better structural accuracy than attempting to summarize and organize in a single pass.

Three observations motivated this decomposition:

Cognitive load reduction. By first identifying atomic concepts without worrying about where they sit in a hierarchy, the LLM can focus entirely on semantic coverage. The organizational pressure is deferred to a second step where the concept inventory is already complete.

Structural fidelity. The second stage receives the extracted concepts as a controlled vocabulary and can devote its full capacity to determining logical connections and multi-level nesting. It does not need to simultaneously decide what is important and how it relates to everything else.

Overcoming linear bias. Most existing tools mirror a document's physical layout, chapter titles become top-level nodes, subsection headers become children, and the result is a shallow copy of the table of contents. EduMap instead employs a recursive mode for longer papers: the system first extracts a high-level framework, then re-enters each branch to extract finer-grained details in parallel. This recursive re-feeding produces a hierarchy based on conceptual relationships rather than page order.

The goal is to turn dense, linear text into a study artifact that supports active engagement rather than passive reading.

3.2 User Interface

EduMap has a simple UI with dual-synchronized view i.e., a Markdown editing panel and an interactive mind map. Both views are powered by a centralized state management system that keeps the text and the visual graph in sync at all times. To prevent the graph from becoming a flat text dump, a dedicated transformer layer automatically calculates the layout so that nodes are positioned logically according to the document's hierarchy.

3.2.1 Core Features

- **Node Management:** Users can directly manipulate the structure by adding nodes through a specialized placement mode, renaming them inline, and editing tags. These modifications automatically flow back into the Markdown source.
- **Expand and Collapse:** The system supports global and level-specific expansion. To manage visual clutter, collapsed subtrees display the count of hidden descendant nodes.
- **Multimodal Ingestion:** The backend processes PDFs to extract figures, formulas, and tables, which are then flattened into a "multimodal context". This allows the LLM to incorporate visual data into the generated mind map.

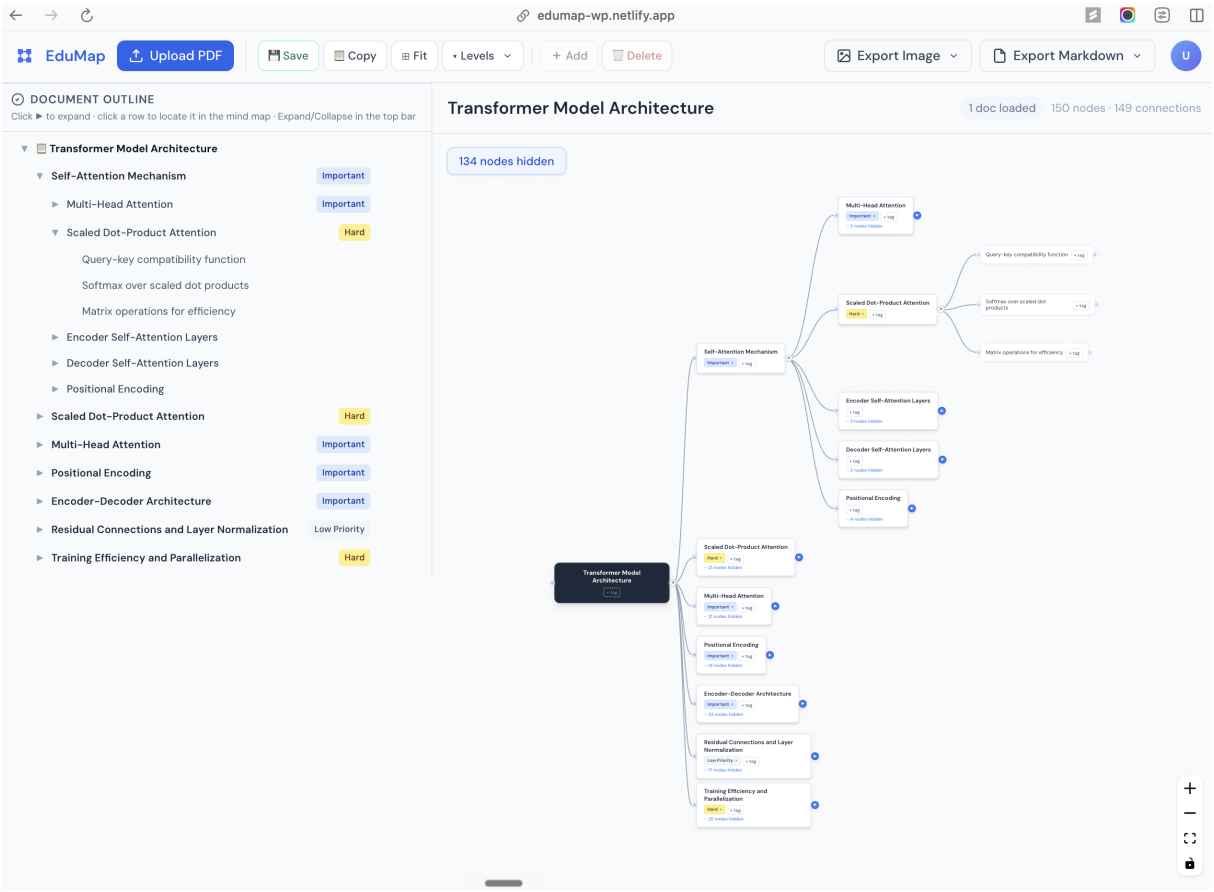

Figure 1: The EduMap interface showing the synchronized Markdown editor (left) and React Flow mind map canvas (right). Nodes display priority tags such as `[Hard]`, and collapsed subtrees show descendant counts.

- **Comprehensive Exports:** Users can export the final product as a PNG image or a Markdown file. The system distinguishes between "Current" edits and the "Original" AI-generated baseline for comparison.

3.2.2 Human-Computer Interaction

We have incorporated the core Human-Computer Interaction (HCI) principles into our application to enhance usability. The important aspects considered include:

- **Direct Manipulation & Feedback:** Expand/collapse actions give instant visual feedback, a pulsing blue ring confirms node selection, and the outline and canvas remain in sync at all times.
- **Visibility of System Status:** Per-node child counts, "subtree hidden" labels, and real-time progress banners keep users informed throughout.
- **Recognition over Recall:** All controls are persistently visible in the toolbar with no hidden menus.
- **Error Prevention & User Control:** Parent deletions require modal confirmation. A bottom undo toaster allows users to reverse accidental edits or deletions.
- **Trust & Minimalism:** A pencil badge appears on any node the user has manually modified, distinguishing human edits from the original AI output (Liao and Vaughan, 2023). The badge is reset only when a new extraction replaces the baseline. A clean white/gray palette with a single blue accent minimizes visual clutter.

3.2.3 Accessibility (WCAG)

EduMap follows WCAG guidelines through semantic HTML5, ARIA roles on modals and dropdowns, and descriptive aria-label attributes on all icon-only buttons. Full keyboard navigation is supported across all toolbar actions, with Escape key

dismissal for modals and programmatic focus updates when nodes are created or searched.

3.3 Backend Features

The backend architecture of EduMap is divided into three decoupled stages, focusing on multimodal content extraction and the structural processing of natural language.

3.3.1 Multimodal PDF Extraction Pipeline

To overcome the limitation of conventional tools that only extract plain text, we deployed a Python-based layout-aware parsing program on the backend. It splits the PDF into four types of content: body text, figures, tables, and formulas. For vector-drawn figures that traditional methods fail to capture, the system directly crops the corresponding page area based on the coordinates of the caption (e.g., "Figure 1").

It also identifies formulas by combining image aspect ratios and text features. After extracting the figures, the system sends the images, along with their surrounding paragraphs from the original paper, to a multimodal model (such as GPT-4o Vision). This allows the model to generate accurate figure descriptions based on the context. Finally, all text and visual content is merged into a multimodal linear context, serving as input data for the next processing step.

3.3.2 Two-Stage Core Concept Extraction

When generating the mind map, to prevent the model from simply copying the physical section headers of the original paper, we adopted a two-step extraction strategy. In the first step (Harvest), the system prompts the Large Language Model (LLM) to extract only core concepts (such as specific methods, principles, or problems) and keeps the list flat.

In the second step (Organize), the system feeds these concepts back into the model, asking it to reorganize the content into a 3- to 6-level tree outline based on logical connections, automatically adding tags like [Hard] or [Important] to complex sections. For longer papers, the system activates a recursive mode: it first extracts the overall framework, then extracts the specific details of each branch in parallel to prevent information loss caused by long texts.

3.3.3 Backend Engineering and State Synchronization

Beyond the core algorithms, the backend provides stable engineering support. The system implements a two-way conversion logic between Markdown and the graph node structure. This ensures that when users edit the sidebar text or drag nodes on the canvas, the data on both sides align in real-time without disrupting the hierarchy.

To ensure system availability, we configured an automatic fail-over mechanism for the model API; if the preferred model service experiences network issues, it automatically switches between Gemini, OpenAI, and Claude. Additionally, combined with the graph layout algorithm and global state management, the backend ensures the smooth operation of the entire multi-view interactive interface.

3.4 Iterative Development Process

The current architecture is the result of significant iteration. Our initial prototype took a naive approach: standard PDF parsing to extract plain text and a single-shot LLM prompt to generate the hierarchical Markdown from the full document.

This pipeline failed in three distinct ways.

Structural hallucination. The single-shot prompt overwhelmed the model's context window. Rather than extracting underlying concepts, the model mirrored the physical section headers of the PDF, producing empty nodes for headings like "Related Work" or "Methodology" without capturing the ideas within them.

Generic visual descriptions. The local BLIP model processed images in complete isolation from the surrounding text. The result was superficial captions (e.g., "a chart with lines") that contributed nothing to the mind map's conceptual structure.

Missing vector graphics. Academic papers frequently use vector-drawn paths for figures rather than embedded raster images. Our initial extractor failed to detect these entirely, leaving gaps in the final mind map wherever a figure should have anchored a structural node.

Each failure pointed toward a specific architectural change. The structural hallucinations drove the shift to the two-stage "Harvest and Organize" strategy. The visual extraction shortcomings motivated the upgrade to context-grounded Vision LLMs and the implementation of caption-anchored region rendering. Taken together, these iterations transformed EduMap from a fragile text summa-

rizer into a multimodal knowledge mapping tool that handles the structural and visual diversity of real academic documents.

4 Results

4.1 Evaluation Metrics

We evaluated EduMap against two baselines: NotebookLM, a commercially available LLM-powered study tool from Google, and a TF-IDF keyword extraction pipeline that serves as a non-neural baseline. Our evaluation paired automated semantic metrics with LLM-based qualitative scoring across a benchmark of five academic papers selected for structural diversity.

Automated Metrics. BERTScore F1 (Zhang et al., 2020) measures semantic similarity at the embedding level between each system's output and the source document. We also computed cosine similarity over TF-IDF vectors to capture surface-level lexical overlap. The two metrics together reveal whether a system is genuinely reorganizing knowledge or simply copying from the source.

LLM-as-a-Judge. Following Zheng et al. (Zheng et al., 2023), we prompted an LLM judge to score each system's output on four dimensions, each on a 0–5 scale: (1) *Conceptual Accuracy*—whether the output reflects key concepts without introducing errors or hallucinations; (2) *Coverage of Key Ideas*—how thoroughly the output captures essential content from the original; (3) *Hierarchical Organization*—the clarity and logic of the structure, and whether it organizes by conceptual relationships rather than mirroring section headings; and (4) *Usefulness for Studying*—how practical the output is for learning, review, or quick comprehension.

Benchmark Dataset. We assembled five open-access papers chosen to stress-test each system under different conditions: AlexNet (Krizhevsky et al., 2012) (9 pages, figure-heavy), Attention Is All You Need (Vaswani et al., 2017) (15 pages, equation-dense), the LLaMA 3 technical report (Dubey et al., 2024) (92 pages), a philosophy of AI paper (12 pages, dense argumentation with few headings), and a quantum game theory paper (28 pages, mathematical proofs and payoff matrices). The rationale was straightforward: a system that only works on neatly structured ML papers is not much use to a graduate student reading across disciplines.

4.2 Quantitative Results

4.2.1 Semantic Preservation vs. Structural Divergence

All three systems landed within a narrow band on BERTScore F1: EduMap at 0.796, TF-IDF at 0.793, and NotebookLM at 0.790 (Figure 2). On its own, this suggests roughly equivalent semantic preservation. The cosine similarity scores tell a different story. TF-IDF averaged 0.959 across papers, which is unsurprising given that it pulls keywords directly from the source with minimal transformation. NotebookLM sat at 0.211, reflecting heavy paraphrasing. EduMap landed at 0.395, between the two.

What matters here is not which system "wins" on cosine similarity but what the gap between the two metrics reveals. High BERTScore paired with low cosine similarity means a system is preserving meaning while substantially restructuring vocabulary, in other words, doing real knowledge reorganization. TF-IDF preserves meaning *and* vocabulary because it barely transforms anything. EduMap preserves meaning while restructuring the text into a hierarchical study aid, which is the behavior we want.

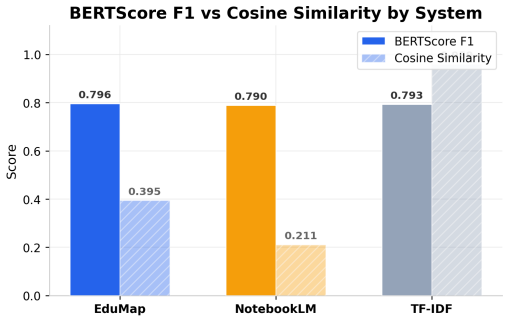

Figure 2: BERTScore F1 vs. Cosine Similarity across systems. All three preserve semantic content to a similar degree, but they differ sharply in how much they restructure the source text.

4.2.2 Educational Quality Across Paper Types

EduMap achieved an overall average judge score of 4.45/5, compared to 4.70 for NotebookLM and 1.15 for TF-IDF. On the surface, NotebookLM edges ahead. But the per-paper breakdown in Figure 3 complicates that reading.

On the Attention and LLaMA 3 papers, EduMap scored 5/5 on Hierarchical Organization. These are technically layered documents with nested concepts (multi-head attention inside encoder-decoder

stacks, or pre-training pipelines feeding into post-training stages), and the mind map format captures those relationships in a way that a flat text summary does not. TF-IDF scored 0 on Hierarchical Organization for the Attention paper, keyword extraction alone cannot recover how concepts relate to one another.

NotebookLM beat EduMap on the Philosophy paper (4.8 vs. 4.2 average). This makes sense: the source material builds a linear argument rather than presenting a hierarchy of concepts, and EduMap's mind map imposed a tree structure where none naturally exists. The Hierarchical Organization score dropped to 3 for that paper specifically. Mind maps are not the right format for every document, and this result reflects that honestly.

On the Quantum paper, NotebookLM scored 5.0 across all four dimensions while EduMap averaged 4.0. The paper's heavy use of Dirac notation and payoff matrices did not pose a problem for NotebookLM's extraction pipeline, but EduMap's "Organize" stage struggled to decompose game-theoretic proofs into a clean concept hierarchy.

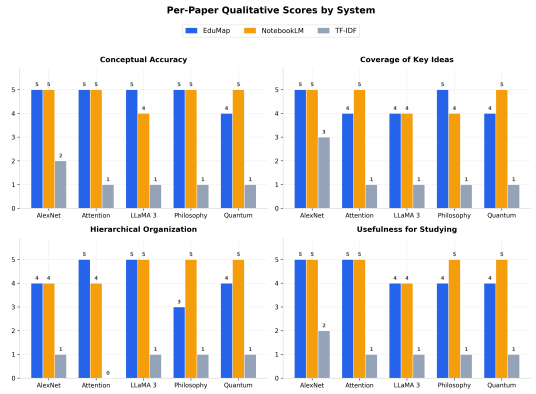

Figure 3: Per-paper qualitative scores across all four evaluation dimensions.

4.2.3 Per-Category Breakdown

Table 1 reports average scores across all five papers for each evaluation dimension. Figure 4 visualizes these averages by paper and system.

Table 1: Average scores per evaluation dimension (0–5 scale). Bold indicates the highest score in each column.

System	Accuracy	Coverage	Hierarchy	Usefulness
EduMap	**4.8**	4.4	4.2	4.4
NotebookLM	**4.8**	**4.6**	**4.6**	**4.8**
TF-IDF	1.2	1.4	0.8	1.0

NotebookLM leads across all four dimensions, with its strongest advantage in Usefulness for Studying (4.8 vs. 4.4). EduMap's weakest dimension remains Hierarchical Organization (4.2), pulled down by the Philosophy paper where a tree structure was a poor fit for argumentative prose. Where EduMap differentiates itself is not in raw scores but in output format: NotebookLM produces flat text summaries, while EduMap produces an interactive, editable mind map that preserves structural relationships between concepts.

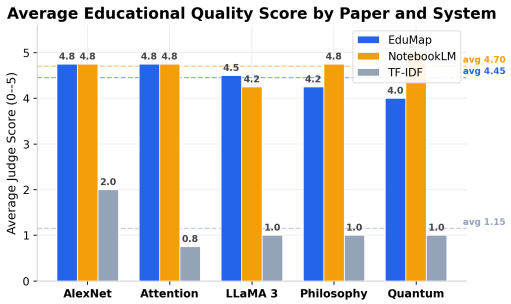

Figure 4: Average educational quality score by paper and system.

4.3 Limitations of the Evaluation

Two caveats apply. First, we did not validate the LLM judge against independent human ratings. Time constraints made a full human evaluation study impractical within the project timeline. Prior work has shown reasonable alignment between LLM judges and human annotators for similar rubric-based tasks (Zheng et al., 2023), but we cannot confirm that holds for our specific rubric and domain. The judge scores should be read as directional rather than definitive. Second, five papers is a small benchmark. We selected them for maximal diversity in length, domain, and structure, but a larger set would strengthen any claims about generalizability. We release the benchmark papers, system outputs, and evaluation scripts to support replication.

4.4 Error Analysis and Mitigating Hallucinations

During initial development, multimodal hallucinations clustered around two areas: complex tables and vector-drawn figures, both of which produced generic or structurally inaccurate descriptions. We addressed this by introducing caption-anchored region rendering and context-grounded prompting strategies into the pipeline. After these changes, qualitative inspection showed a substantial reduction in inaccuracies. Vision model outputs shifted

from superficial visual summaries (e.g., “a chart with lines”) to precise explanations aligned with the authors’ intent. The successful capture of vector-drawn figures also eliminated the missing-node errors that had previously left gaps in the mind maps.

These architectural improvements resolved the primary multimodal extraction errors, but a closer analysis of our test set reveals two remaining failure modes rooted in LLM orchestration rather than visual extraction.

Deep hierarchy flattening. In the LLaMA 3 paper, which contains deeply nested hyperparameter ablations, EduMap occasionally flattened Level-4 headings into Level-3 sibling nodes. The primary tree structure was captured correctly, but the deepest branches were prematurely capped. For comparison, NotebookLM failed to capture this nested structure at all. The failure occurs during the “Organize” stage: when the LLM processes over 100 atomic concepts simultaneously, context window limitations force the model to compress deeper branches to conserve output tokens. A natural fix is to trigger the recursive extraction mode more aggressively for deeply nested sections, forcing the LLM to process sub-trees independently rather than fitting the entire hierarchy into a single inference call.

Complex table misalignment. The second failure mode involves cross-page multimodal elements. In the Quantum paper, a data table spanning two pages was misextracted. EduMap’s vision pipeline captured the raw text, unlike standard text extractors, which produced garbled strings, but the hierarchical relationship of merged column headers was lost. Because the layout-aware pipeline evaluates bounding boxes strictly on a per-page basis, the vision model processed two separate images without understanding their shared columnar structure, producing disjointed Markdown. Implementing a cross-page bounding box stitching heuristic, or integrating a dedicated Table Structure Recognition (TSR) model, would address this spatial limitation.

5 Discussion

5.1 Significance and Novelty

As evidenced above, multimodal inputs play an important role in increasing coverage. The primary innovation of EduMap revolves around moving away from the static AI-generated artifact towards personalized "study artifact," which reflects the idea that mind maps should be viewed as actively editable artifacts of thought. Unlike existing systems, EduMap employs a unique synchronization mechanism that allows users to modify the created knowledge structure, thus creating truly personal outputs rather than fixed summaries. Thus, the goal of EduMap is not to replace understanding but to serve as scaffolding for constructing knowledge.

5.2 Replicability and Datasets

Our full codebase, including installation instructions, prompt templates for both the Harvest and Organize stages, and model configuration files, is publicly available on GitHub. The evaluation benchmark, comprising the five source PDFs, each system’s generated outputs, ground-truth hierarchies, and the scoring scripts used for both BERTScore and LLM-as-a-Judge evaluation, is included in the repository. The pipeline relies on publicly available models (LiLTv2 for layout understanding, GPT-4o for visual captioning) and standard LLM APIs, with no proprietary training data or fine-tuned weights required to reproduce our results.

5.3 Ethics and Limitations

The main ethical issue associated with EduMap involves the potential for over-reliance on machine-generated content, especially regarding technical or factual content wherein hallucinations could harm the user’s understanding. To address this issue, the human-AI partnership approach is adopted. The design of the interface encourages active engagement with content in order to edit and reorganize it; in other words, the AI-generated content is treated merely as a preliminary step in the generation process.

A secondary concern involves data privacy. EduMap’s pipeline routes uploaded PDFs through commercial APIs such as GPT-4o and Gemini for figure captioning and concept extraction. For users working with unpublished manuscripts, grant proposals, or proprietary course materials, this means sensitive content leaves the local environment and is processed by third-party servers. While the current deployment relies on these services for quality, a production-ready version would need to offer a local inference option or provide clear disclosure about which data is transmitted externally.

In terms of technical aspects, the key limitation involves reliance on the functioning of the backend server, as well as having a properly configured API key for connecting to it. Moreover, the multimodal

pipeline has been optimized and improved, but the system remains susceptible to the occasional multimodal or table hallucinations.

6 Conclusions

EduMap successfully transforms dense academic PDFs into interactive, hierarchical mind maps. By combining multimodal extraction with user-centered editing, we provide a significant improvement over linear summarization tools. A core engineering contribution of this project is the development of a paired markdown and graph editing system that provides users with an exportable original baseline for comparison. Future extensions include citation analysis to highlight the most impactful research within a document.

7 Future Enhancements

Several directions for future work could strengthen EduMap's utility and reliability. Fine-tuning the underlying models on domain-specific corpora from fields such as medicine, law, or engineering would improve the precision of concept extraction in specialized disciplines where general-purpose models tend to flatten terminology. Linking individual nodes to external learning resources, including academic papers, videos, and formative assessments, would help students bridge gaps between topics represented in the map.

The system would also benefit from a confidence-signaling layer that flags nodes generated from complex tables or figures when the model's certainty is low, reducing the risk of students relying on inaccurate output. On the interaction side, enabling real-time collaborative editing would open the tool to group study and classroom use, while optimizing the interface for touch-based devices would broaden accessibility to tablets and smartphones. Finally, implementing version history would allow users to track changes and revisit earlier states of their mind maps, a feature particularly valuable in long-term study contexts where understanding evolves over time.

8 Acknowledgments

We thank Professor Andy Exley and Daanish Hindustani for their invaluable input, guidance, and support throughout the development of this project.

We would also like to express our gratitude to our fellow students for their insightful recommendations and constructive feedback provided during the initial idea pitch and the final poster presentation, which significantly helped refine our system's core features.

9 References

References

Maryam Azam, Shah Khalid, Sulaiman Almutairi, and Hasan Ali Khattak. 2025. Current trends and advances in extractive text summarization: A comprehensive review. *IEEE Access*, 13:28150–28166.

Gaye Colakoglu, Gürkan Solmaz, and Jonathan Fürst. 2026. Problem solved? information extraction design space for layout-rich documents using LLMs. *arXiv preprint arXiv:2502.18179*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Wenjun Ke, Yifan Zheng, Yining Li, and Hengyuan Xu. 2026. Large language models in document intelligence: A comprehensive survey, recent advances, challenges, and future trends. *ACM Transactions on Information Systems*, 44:1–64.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25.

Q. Vera Liao and Jennifer Wortman Vaughan. 2023. Ai transparency in the age of llms: A human-centered research roadmap. *arXiv preprint arXiv:2306.01941*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30.

Jiapeng Wang, Zening Lin, Dayi Huang, Longfei Xiong, and L. W. Jin. 2025. LiLTv2: Language-substitutable layout-image transformer for visual information extraction. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 21:1–27.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36.