2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Paper ID #52995

Identifying, categorizing, and evaluating the taxonomy of STEM content and the obstacles in converting handwritten lecture materials of engineering courses for improved accessibility

Advita Gelli, University of Illinois at Urbana - Champaign
Alan Tao, University of Illinois at Urbana - Champaign
YANGYANG ZHANG, University of Illinois at Urbana - Champaign
Nikitha Sundaram
Sonika Tamilarasan, The University of Illinois at Chicago
Jonathan Hogg, University of Illinois at Urbana - Champaign
Yang Victoria Shao, University of Illinois at Urbana - Champaign

Yang V. Shao is a Teaching Associate Professor in electrical and computer engineering department at University of Illinois Urbana-Champaign (UIUC). She earned her Ph.D. degrees in electrical engineering from Chinese Academy of Sciences, China. She has worked with University of New Mexico before joining UIUC where she developed some graduate courses on Electromagnetics. Dr. Shao has research interests in curriculum development, assessment, student retention and student success in engineering, developing innovative ways of merging engineering fundamentals and research applications.

Dr. Chrysafis Vogiatzis, University of Illinois at Urbana - Champaign

Dr. Chrysafis Vogiatzis is a teaching associate professor for the Department of Industrial and Enterprise Systems Engineering at the University of Illinois Urbana-Champaign. Prior to that, Dr. Vogiatzis was an assistant professor at North Carolina Agricultural and Technical State University. His current research interests lie in network optimization and combinatorial optimization, along with their vast applications in modern socio-technical and biological systems. He is serving as the faculty advisor of the Institute of Industrial and Systems Engineers, and was awarded the 2019, 2023, and 2024 Faculty Advisor award for the North-Central region of IISE. Dr. Vogiatzis was awarded ASEE IL/IN Teacher of the Year in 2023.

Dr. Pablo D Robles-Granda, University of Illinois at Urbana - Champaign

Pablo Robles-Granda is a Teaching Assistant Professor at the University of Illinois at Urbana-Champaign.

Prof. Lawrence Angrave, University of Illinois at Urbana - Champaign

Dr. Lawrence Angrave is an award-winning computer science Teaching Professor at the University of Illinois Urbana-Champaign. He creates and researches new opportunities for accessible and inclusive equitable education.

Dr. Hongye Liu, University of Illinois at Urbana - Champaign

Hongye Liu is a Teaching Assistant Professor in the Dept. of Computer Science in UIUC. She is interested in education research to help students with disability and broaden participation in computer science.

Identifying, categorizing, and evaluating the taxonomy of STEM content and the obstacles in converting handwritten lecture materials of engineering courses for improved accessibility

Abstract

The digital accessibility of handwritten STEM materials remains a barrier to inclusive engineering education, particularly for students who rely on screen readers and text-to-speech technologies. The goal of this study is to better understand the challenges involved in converting handwritten mathematical and other STEM content into structured digital formats. We present a comprehensive evaluation of the current state of Handwritten Mathematical Expression Recognition (HMER) models, with a focused analysis of the MinerU-2.5 Vision Language Model's performance. By analyzing the equations from multimodal engineering and computing lecture slides, we identify a critical performance gap between symbolic recognition and structural representation. Our findings show that while the model achieves a high rate (87.0%) in recovering mathematical meaning, spatial failures such as collapsed formatting, misinterpreted sub/superscripts, and dropped summation bounds result in a lower overall interpretability rate (72.4%). We formalize these challenges into a three-part taxonomy of obstacles to symbolic accuracy: structural and spatial failures in preserving 2D arrangement or delimiters, environmental interference from multimodal elements like arrows or overlapping boundaries, and semantic inaccuracies as a result of ambiguous character substitutions. Furthermore, our layout analysis reveals a 66.5% accuracy rate for page-level decomposition, with a 9.6% failure rate primarily caused by multimodal complexity. These obstacles are also formalized into three categories based on whether they stem from issues with type-casting, environmental boundaries, or hierarchical structure. The paper concludes that modern HMER systems are currently bottlenecked more by spatial hierarchy modeling than raw symbol detection. The classification of obstacles in converting handwritten mathematical and other STEM content in engineering courses to LaTeX helps increase the digital accessibility of STEM resources by providing a roadmap for developing practical guidelines for instruction design, benchmarking, and tool development.

Introduction

Motivation for digitizing slides

Digital accessibility of STEM content remains a significant challenge in higher education, particularly regarding handwritten materials. Lecture slides, often the primary medium for presenting course material, are commonly unstructured and multimodal. The material often integrates typed text and symbolic equations with spontaneous handwritten annotations that provide visualizations and context for step-to-step problem-solving. While this handwritten

content is visually accessible during a live or recorded lecture, it presents major challenges for digital accessibility. Lecture slides are often distributed to students as images or PDFs, existing tools and technologies, such as text-to-speech (TTS) and screen readers, struggle to recognize, interpret, and represent the underlying information consistently and within the context of the original lecture/recording. Additionally, prior research has demonstrated the difficulty in transcribing multimodal math content [1].

Beyond these technical obstacles, previous research under the Universal Design for Learning (UDL) highlighted a systemic accessibility gap in STEM materials [2]. An additional UDL study demonstrated that providing digital alternatives, such as a compilation of transcriptions from lecture videos, significantly improves the perceived learning for students with disabilities (SWD) and students with accessibility needs (SWAN) [3]. Yet, a critical need to make mathematical equations, terms, and diagrams accessible persists. By digitizing STEM content into machine-readable formats (eg: LaTeX or Markdown), institutions can fulfill UDL principles of providing multiple means of representation, enabling screen readers to interpret complex notation, and allowing students to engage with the material without the risk of transcription errors.

The urgency of this need is further reflected in localized student data. Internal surveys conducted across 2 computer science courses, 1 industrial engineering, and 1 electrical engineering course at the University of Illinois Urbana-Champaign indicate a high demand for digitized alternatives to traditional handwriting in lecture content. In our companion paper, based on a study involving 236 engineering students, 59.3% of respondents preferred handwritten text to be supplemented in a typed form. This demand is even more pronounced for mathematical equations, where 70.8% of respondents highlighted handwritten equations as a priority for LaTeX or Markdown conversion. Students consistently selected handwritten content as the least accessible element in their STEM courses. These findings underscore the necessity for high-fidelity conversion tools that can bridge the gap between spontaneous handwritten instruction and structured digital formats.

Purpose

This work is specifically motivated by the need to understand the underlying structure and complexity of multimodal STEM materials. To address this, we evaluate the efficacy of multiple Handwritten Mathematical Expression Recognition (HMER) models in digitizing complex engineering lecture slides with a focus on MinerU-2.5, a modern document-to-content parser [4]. We propose a bimodal evaluation framework to provide a comprehensive assessment: a top-down approach that analyzes ‘high-risk’ syntax and software constraints inherent in LaTeX conversion alongside a bottom-up approach, which utilizes a dataset of slides from the University of Illinois Urbana-Champaign to test raw symbolic and spatial-detection accuracies.

By consolidating these findings, this paper aims to formalize a taxonomy of mathematical symbols designed to support automated context refinement and transcription that currently remain underdeveloped. After identifying the specific categories of obstacles that hinder accurate conversion of handwritten mathematical content, we outline a roadmap for both developers building transcription tools and instructors designing course materials to maximize digital accessibility.

Literature and Background

Top-down approach

The top-down analysis investigates HMER performance by examining the formal software and syntactical frameworks that govern how visual representations are translated into executable LaTeX code. Unlike bottom-up analysis, this approach treats LaTeX as a structured programming and markup language rather than a natural language string.

(i) Software Constraints

The conversion of handwritten math into LaTeX is constrained by the target execution environment. Generated code must satisfy the formal requirements of LaTeX compilers and specific package ecosystems. Even when a model correctly identifies a symbol, the transcription fails if the HMER system does not have the capability to automatically import or support the necessary packages (e.g., amsmath) [5].

Furthermore, lecture materials often rely on custom macros (def, newcommand) that are not standardized, presenting a significant barrier to reliable rendering [6]. Even if a model correctly recognizes the intent of a symbol, the transcription fails if the output cannot be interpreted by the compiler. This creates the need for a conversion pipeline that accounts for ecosystem validity, ensuring that the recognized output is not just a visual match but also able to be compiled.

(ii) 'High-Risk' Syntax and Ambiguity

Handwritten STEM content is inherently prone to misinterpretation due to its 2D nature and the overlap of symbols. Utilizing the LaTeX Wikibook standards, we identify "high-risk" syntactic components where density and spatial arrangement create uncertainty. These include advanced structures such as Matrix Environments, Vertical Alignments for multi-line derivations, or delimiters and nested fractions.

Linguistic research suggests that mathematical expressions exhibit "hierarchical, recursive structures" similar to natural language [7]. However, where natural language utilizes binary branching, mathematical notation contains ternary structures (e.g., $3 + 5$) that create practical issues for linearization. Resolving these ambiguities is difficult due to a lack of explicit symbol-to-trace alignment.

Traditional encoder-decoder models often "guess" the identity of a symbol based on attention weights, which can misalign under style variability. To that end, object detection-driven approaches enforce clear instance boundaries before modeling spatial relationships [8].

(iii) Structural Dependencies and Hallucinations

A critical failure mode in modern HMER is the generation of "hallucinated" LaTeX syntax, where outputs delivered with high linguistic confidence are actually invalid. Evidence suggests that hallucinations in structured languages are systemic and repeatable as models fail in the same way for structural patterns, suggesting these are stable failures [5].

In STEM contexts, these hallucinations often break long-range structural dependencies, such as pairs of delimiters or nested environments. Sequence-based models struggle with hierarchical constraints that span across the entire expression [9]. Since mathematical expressions rely on the preservation of the 2D layout, even a single relation error such as misinterpreting a superscript as

a standard character can cause a series of downstream transcription failures [10].

(iv) Limitations of Syntax Trees and Cognitive Hierarchies

Cognitive psychology treats math as a subset of linguistic syntax with its own grammar and structure. Recent theories argue that mathematical expressions are processed in a way similar to sentences: by breaking them into a hierarchy of pairs to form a syntax tree [7]. While many HMER models utilize syntax-tree based validation to ensure grammatically-correct LaTeX, this approach has its own limitations. Syntactic validity does not guarantee visual correctness as this method only ensures that the output is internally consistent with balanced braces and legal characters. A LaTeX string could be balanced and valid according to a context-free grammar while rendering an incorrect segmentation of the original handwritten strokes [11].

Furthermore, as the depth of a syntax tree increases, which is common when transcribing STEM content, the potential branching paths can exceed a model's search capacity and lead to a degradation in recognition quality.

(v) Modality of the Input Data

"Online" ink (stroke sequences) and "offline" images (pixels) have different structural constraints. Online data provides temporal information about the order and trajectory of strokes, which is incredibly useful for segmenting overlapping symbols and identifying the intended flow of a derivation [10]. Foundational models like InkFM have great accuracy due to their ability to interpret the temporal data and "see" the construction process of a formula [9]. This suggests that top-down errors in offline systems might arise from the loss of this temporal hierarchy. Relying purely on spatial density might be insufficient for resolving complex, multi-layered equations common in engineering notes.

(vi) Incorrect Type-Casting

Before most models recognize individual symbols from equations, they perform a page-level decomposition to decide if a region is an equation, a table, a diagram, or a simple annotation. Errors often occur before the first symbol is even read if a system misidentifies a math region as a sketch or an enclosure [9] as the classification acts as a type-casting mechanism for the decoder to adjust its objective for LaTeX tokens or standard text. If a system lacks accurate information, it may attempt to parse a red arrow or a "cross-out" (an 'X') as a mathematical symbol like downarrow or times, leading to environmental interference errors.

Methodology

Model Selection

The field of HMER has evolved from multi-stage pipelines and neural encoder-decoder models to unified context-aware Vision-Language Models (VLMs). We categorize the existing landscape of HMER technologies and evaluate their suitability for digitizing complex engineering lecture slides in Appendix A.

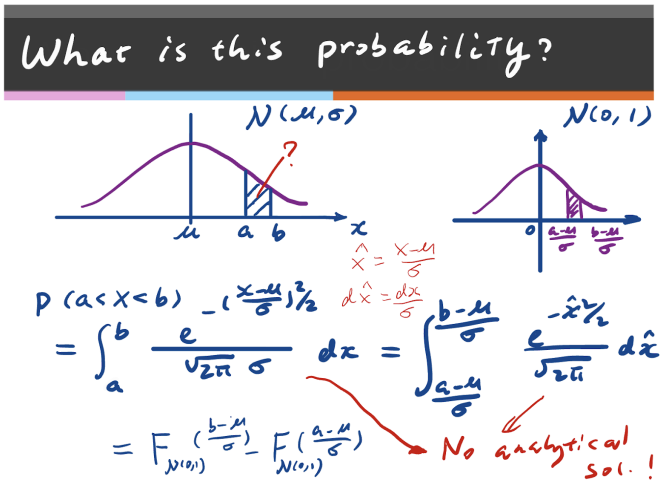
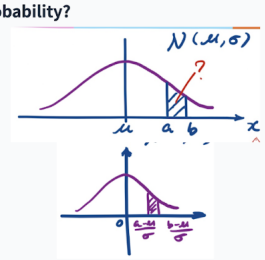
Index	Lecture Slides vs OCR Outputs
1	Kullback-Leibler (KL) Divergence Retrieval Model Query Likelihood $f(q,d) = \sum_{w \in d, w \in q} c(w,q)[\log \frac{p_{Seen}(w \mid d)}{\alpha_d p(w \mid C)}] + n \log \alpha_d$ KL-divergence (cross entropy) $f(q,d) = \sum_{w \in d, p(w \mid \theta_Q) > 0} [p(w \mid \hat{\theta}_Q)] \log \frac{p_{seen}(w \mid d)}{\alpha_d p(w \mid C)} + \log \alpha_d$ Query LM $p(w \mid \hat{\theta}_Q) = \frac{c(w,Q)}{\mid Q \mid}$
	MinerU-2.5 Output **Kullback-Leibler (KL) Divergence Retrieval Model** Query Likelihood $f(q,d) = \sum_{w \in d} \sqrt{c(w,q)} [\log \frac{p_{seen}(w \mid d)}{\alpha_d p(w \mid C)}] + n \log \alpha_d$ KL-divergence (cross entropy) $f(q,d) = \sum_{w \in d, p(w \mid \theta_Q) > 0} [p(w \mid \hat{\theta}_Q)] \log \frac{p_{seen}(w \mid d)}{\alpha_d p(w \mid C)} + \log \alpha_d$ Query LM $p(w \mid \hat{\theta}_Q) = \frac{c(w,Q)}{\mid Q \mid}$
	PaddleOCR-VL Output Kullback-Leibler (KL) Divergence Retrieval Model Query Likelihood KL-divergence (cross entropy) $f(q,d) = \sum_{w \in d, w \in q} c(w,q) [\log \frac{p_{Seen}(w \mid d)}{\alpha_d p(w \mid C)}] + n \log \alpha_d$ $f(q,d) = \sum_{w \in d, p(w \mid \theta_Q) > 0} [p(w \mid \hat{\theta}_Q)] \log \frac{p_{seen}(w \mid d)}{\alpha_d p(w \mid C)} + \log \alpha_d$ Query LM $p(w \mid \hat{\theta}_Q) = \frac{c(w,Q)}{\mid Q \mid}$
2	
	MinerU-2.5 Output $p(a < x < b) - (\frac{x-a}{\sigma})^2/2$ $d\hat{x} = \frac{dx}{\sigma}$ $= \int_a^b \frac{e}{\sqrt{2\pi}\sigma}$ $\frac{dx}{d\sigma} = \int_{\frac{a-4}{\sigma}}^{\frac{b-4}{\sigma}} \frac{e^{-\hat{x}^2/2}}{\sqrt{2\pi}} d\hat{x}$ $= F_{N(0,1)}(\frac{b-\mu}{\sigma}) - F_{N(0,1)}(\frac{a-\mu}{\sigma})$
	PaddleOCR-VL Output **What is this probability?**  $P(a < x < b) = (\frac{x-\mu}{\sigma})^2 \frac{e}{\sqrt{2\pi}\sigma} dx = \frac{b-\mu}{\sigma} \frac{e}{\sqrt{2\pi}} d$

Table 1: MinerU-2.5 v/s PaddleOCR-VL Markdown Rendering

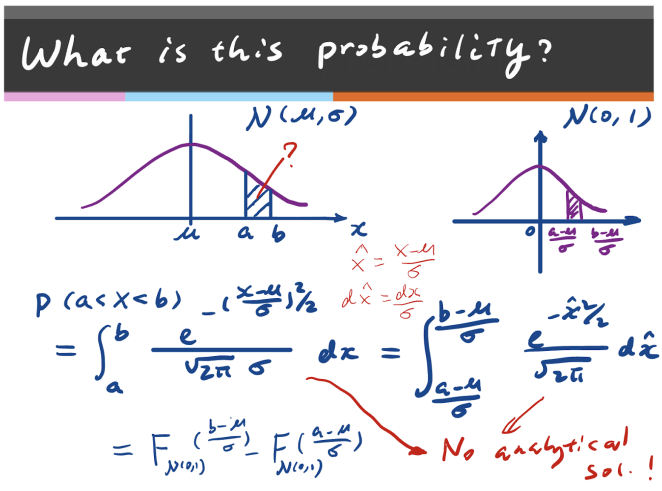
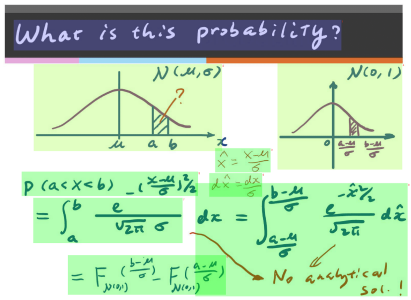
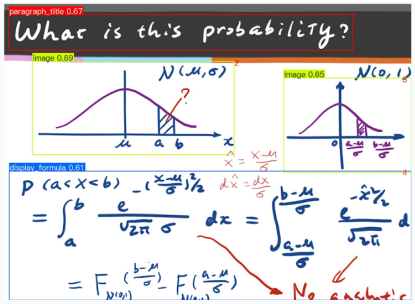
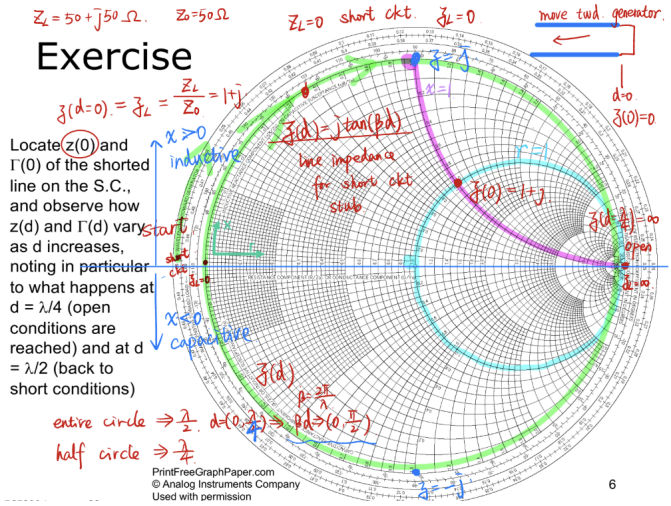
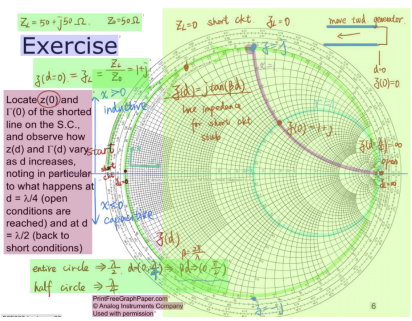
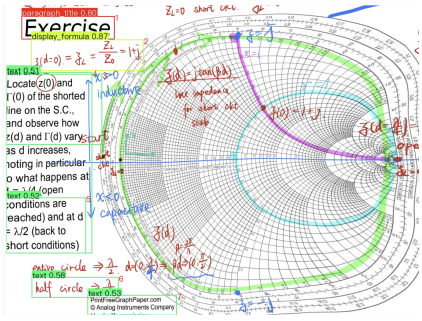
<table>
<tr><th>Index</th><th>Lecture Slides vs OCR Outputs</th></tr>
<tr><td>1</td><td> 
</td></tr>
<tr><td>MinerU-2.5 Output</td><td>PaddleOCR-VL Output</td></tr>
<tr><td>  </td><td>  </td></tr>
<tr><td>2</td><td> 
</td></tr>
<tr><td>MinerU-2.5 Output</td><td>PaddleOCR-VL Output</td></tr>
<tr><td>  </td><td>  </td></tr>
</table>

Table 2: MinerU-2.5 v/s PaddleOCR-VL Layout Detection

We selected MinerU-2.5 as the primary tool for this research due to its 'Decoupled Architecture' and adaptability to high-density engineering inputs. By breaking complex formulas into lines and recombining them, MinerU overcomes the common limitation of VLMs on long mathematical sequences, enabling it to outperform general-purpose AI models like GPT-4o in STEM document parsing precision.

In contrast, we also evaluated TexTeller, a model built on scaling laws. While it achieves SOTA precision on clean symbols by training on the massive Tex80M dataset [12], its reliance on a standard Vision Transformer (ViT) encoder makes it less resilient at handling real-world noise.

On the other hand, PaddleOCR-VL recently surpassed MinerU-2.5 on the OmniDocBench v1.5 benchmark, recording the highest Formula-CDM score (91.22) and the lowest Text-Edit Distance (0.035). Its PP-DocLayoutV2 engine and ability to process images at their native aspect ratio make it suitable for preserving the syntax of handwritten subscripts in STEM contexts [13].

In our evaluation of complex engineering slides, we observed a performance trade-off: while PaddleOCR-VL excelled in raw syntax accuracy for text-heavy slides, its performance diminished on layouts featuring "messy" handwritten annotations and overlapping images. For instance, the first entry in Table 1 shows a structured slide with complex mathematical notation that PaddleOCR-VL was able to transcribe more accurately than MinerU. However, the second entry features a slide whose layout contains hand-drawn arrows and multiline equations. PaddleOCR-VL was only able to detect one equation, which was degraded due to the annotations.

In the highly unstructured scenarios, MinerU's layout-first approach performed significantly better at detecting the presence of equations as shown in Table 2, effectively solving the 'high density' problem in engineering slides, where math and text are tightly combined. Additional examples of the model outputs can be found in Appendix B.

Bottom-up approach

To examine the challenges involved in converting multimodal mathematical content from engineering lecture slides into structured digital representations, we adopt a bottom-up evaluation approach that analyzes performance at the level of individual equations and overall layout separately.

For the symbolic accuracy evaluation, rather than evaluating entire slides as a single unit, our analysis treats each instructional mathematical expression independently. This design allows us to isolate fine-grained recognition failures that are often obscured by aggregate accuracy metrics. This approach is particularly important for STEM materials, where mathematical meaning is conveyed not only through symbols, but also through spatial arrangement of variables, alignment, and contextual structure.

In instructional settings, errors in layout or structure can substantially reduce accessibility and interpretability, even when individual symbols are correctly recognized. By focusing on equations and layout separately, we aim to distinguish between symbolic correctness, structural fidelity, and detection failures, and to better understand how each one contributes to overall accessibility challenges.

Dataset Composition

The empirical basis for this study consists of multimodal lecture slides drawn from engineering and computing courses at the University of Illinois Urbana-Champaign. The four lectures analyzed in this study cover topics in statistics, information systems, industrial engineering, and electromagnetics. These courses were selected to capture a range of notation styles, equation densities, and instructional practices commonly found in STEM education.

Symbolic Scoring Rubric

To ensure a consistent evaluation of equations across these varied lectures, we first established a standardized unit of analysis. Each lecture slide was manually reviewed to identify instructional mathematical content. We define a "ground truth instance" (GTI) as any mathematical expression that contributes directly to the explanation of course concepts, including standalone equations, multi-line derivations, symbolic definitions, and expressions integrated with brief textual annotations. Decorative symbols, page numbers, and title-only slides were excluded from analysis.

With these GTIs identified, we compare the original content to the corresponding LaTeX output generated by the MinerU-2.5 model. Performance is assessed along three complementary dimensions: mathematical correctness, spatial fidelity, and exclusion. This multi-dimensional evaluation framework is designed to capture not only whether an equation is recognized, but also whether the recognized output preserves the instructional intent of the original content.

(i) Mathematical correctness

This is evaluated based on whether the predicted LaTeX output preserves the mathematical meaning of the ground truth equation. Each equation is assigned a 'Math Score' on a four-point ordinal scale ranging from 0 to 3. A score of 3 indicates that the predicted expression is mathematically and structurally identical to the ground truth. A score of 2 corresponds to minor symbol-level or formatting errors, such as spacing inconsistencies or missing diacritics, that do not alter the underlying mathematical meaning. A score of 1 reflects structural errors that distort or obscure meaning, while a score of 0 indicates complete failure or exclusion. For aggregate analysis (see: Results, Section A), equations receiving a Math Score of 2 or 3 are classified as mathematically correct, reflecting a practical notion of correctness aligned with instructional usability.

(ii) Spatial Fidelity

In addition to symbolic accuracy, we explicitly evaluate whether the predicted output preserves the spatial arrangement of the ground truth instance (GTI). Handwritten mathematics frequently relies on spatial organization to encode meaning, particularly for multi-line derivations, summations, integrals, and expressions with alignment constraints. Common spatial failures include missing bounds on summations or integrals, improper separation of multi-line equations, loss of alignment across derivation steps, and incorrect grouping of symbols due to spacing or line breaks. An equation is classified as 'spatial arrangement incorrect, but math correct' if its mathematical meaning is preserved, but its spatial organization is degraded in a way that reduces readability or instructional clarity.

(iii) Exclusion Errors

Finally, we identify exclusion errors, which occur when a valid ground truth instance is present in the lecture slides, but the model produces no corresponding mathematical output. These cases include empty outputs, non-mathematical text, or instances where equations are entirely skipped. Components that were intentionally excluded from evaluation, such as title-only slides or non-instructional content, are not counted as exclusions. Tracking exclusions separately allows us to distinguish detection failures from hard-to-convert equations.

Using these criteria, we compute symbolic aggregate metrics for each course, including the total number of GTIs, the number of mathematically correct equations, the number of equations that are mathematically correct but have an incorrect layout, and the number of complete exclusions.

Layout Scoring Rubric

To supplement the symbolic analysis, we developed a secondary rubric to evaluate the model's performance in page-level decomposition and region classification. Layout detection serves as a critical first step in the conversion pipeline as a misidentified region could lead the decoder to attempt to parse, say, a diagram as LaTeX tokens. Our scoring system is also designed to quantify how effectively models isolate these functional regions without triggering error propagation, where a missed or misclassified bounding box leads to the total loss of mathematical content in later stages.

(i) Merge Definitions

To standardize our qualitative findings into quantitative data, we define merges as follows:

- Minor Merge: The consolidation of two distinct ground-truth instances into a single bounding box. While this may cause reading order issues, the symbolic content is often still recoverable.
- Major Merge: The consolidation of three or more distinct ground-truth instances. This typically occurs in high-density STEM slides where math and text are tightly integrated, resulting in formatting that renders the output uninterpretable.

(ii) Region Classification (Color Guide)

Classification accuracy is the most vital component of layout detection, as it dictates the processing of the content in later steps. We utilized the following color-coded system from the MinerU-2.5VL model to categorize the multimodal elements found in engineering slides:

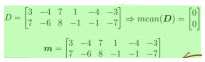
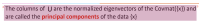
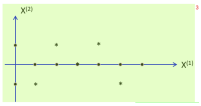
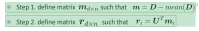
Light Green	Red	Greenish-Yellow	Dark Green	Blue/Purple
				
Mathematical Expressions	Standard Instructional Text	Images, Graphs, or Diagrams	Lists and Bulleted Points	Titles, Headers, and Captions

Table 3: MinerU-2.5 Layout Detection Color Guide

(iii) Scoring Scale and Logic

We utilize a four-point ordinal scale (0–3) to measure the fidelity of detected bounding boxes. This choice allows us to distinguish between minor aesthetic overlaps and major structural failures that would break the reading order or mathematical meaning for a student. To that end, we established a hierarchy of importance: Color/Classification $>$ Merge $>$ Split. This prioritization reflects the reality that a misclassified region causes more significant environmental interference than a split box.

- Score 3 (Perfect): The bounding box accurately encompasses the intended element with the correct classification color.
- Score 2 (Minor Error): The element is identified but contains a ”Minor Merge” (combining two distinct elements into one), a split, or includes unnecessary ”noise” from the multi-modal background.
- Score 1 (Major Error): The detection contains a ”Major Merge” (combining more than two elements), uses an incorrect classification color, or omits essential instructional information.
- Score 0 (Complete Failure): The model fails to detect the region entirely or produces an output that does not correspond to the ground-truth layout.

Together, the symbolic and layout scoring rubrics provide a quantitative summary of recognition performance while preserving interpretability at the equation level. The methodology also reveals limitations that are not captured by string-based similarity metrics alone and serves as an empirical foundation for the taxonomy of recognition obstacles presented in the Results section.

Results (Taxonomy of Obstacles)

The evaluation of MinerU-2.5 across 5 lectures and 292 ground truth instances reveals a complex dependence between symbolic recognition and document structure preservation. To quantify the challenges of interpreting both the functional hierarchy and multimodal environment of a lecture slide, we analyze the performance of the model by calculating the symbolic and structural accuracies separately.

Structural Obstacles (Symbolic Accuracy Analysis)

Performance Metrics Aggregates

We utilize two primary metrics to assess model efficacy at maintaining symbolic accuracy: Strict Instance Rate (Strict InstRate) requires both symbolic and structural perfection only counts outputs with scores of 3 (Fully correct), whereas Lenient Instance Rate (Lenient InstRate) considers a rendered output successful if its mathematical meaning is recoverable despite spatial arrangement inaccuracies and accounts for outputs with scores of either 2 or 3 (Math correct). These two metrics are represented as percentages and allow us to distinguish between significant failures and minor formatting errors. Furthermore, we have included a count of the GTIs that were excluded by the model in its final output to provide a transparent assessment of its capabilities in converting STEM lectures to LaTeX.

Note that non-instructional content (eg. title slide, page numbers, etc) do not correspond to GTIs and are therefore excluded from evaluation. Furthermore, GTIs from slides that were entirely omitted by the model were also disregarded from this evaluation, as we believe a qualitative analysis (see: Results, Section C) of these slides would be more informative.

Course	**Total GTIs**	**Fully correct**	**Math correct**	**Excluded GTIs**	**Strict InstrRate**	**Lenient InstrRate**
Statistics	159	106	120	21	66.7%	75.5%
Information Systems	57	50	57	0	87.7%	100%
Electromagnetics	76	60	71	4	78.9%	93.4%
Industrial Engineering	77	51	73	4	66.2%	94.8%
Total	**369**	**267**	**321**	**29**	**72.4%**	**87.0%**

Table 4: Symbolic Performance Metrics Aggregates of MinerU-2.5 Across 4 STEM Courses

As shown in Table 4, MinerU-2.5 demonstrated varying degrees of efficacy across the four STEM courses. While it achieved a high Lenient InstRate of 100% in the Information Systems Course, its performance was more challenged in Statistics and Industrial Engineering, where the Strict InstRate dropped to 66.7% and 66.2% respectively. This could be attributed to the Statistics course's higher multimodal complexity, which includes diagrams, handwritten and typed equations, text, and annotations, in contrast to the Information Systems course, which primarily contained text and typed equations.

Overall, the data shows an average gap of 16.1% between Strict and Lenient InstRates across all courses. The gap is particularly evident in Industrial Engineering which showed a 28.6% difference, likely due to a smaller number of GTIs. The findings suggest that while the model often captures the mathematical details, it frequently struggles with how they should be presented.

Taxonomy of Symbolic Obstacles

Based on a qualitative analysis of the "Math Score" for GTIs from non-omitted slides, we have identified three primary categories of obstacles that might hinder high-fidelity HMER.

(i) Structural and Spatial Obstacles

These obstacles occur when the model correctly identifies individual symbols but fails to preserve the details of its visualization. This category represents the primary reason behind the performance gap of Strict and Lenient InstRates noted in Table 3, as spatial arrangement failures often degrade the LaTeX output without destroying the underlying symbolic structures.

- Grouping and Spacing: In handwritten contexts, whitespace often works as a delimiter. In the Statistics course, several instances occurred where multiple spaces between independent parts of an equation were collapsed in the output, leading to a single, unreadable LaTeX string.
- Subscript and Superscript: Models often struggle with annotated words and phrases. In the Electromagnetic course, the model rendered a_y as ay and failed to recognize tildes as belonging to the base character, instead treating them as stray marks or separate symbols.
- Summation and Integral Bound: A recurring failure in the Information Systems course involved dropping the limits for summations. While the Sigma and body were captured, the bounds (e.g. i=1 to n) were either omitted or rendered as standard subscripts. This could lead to a loss in the interpretability by screen readers.

(ii) Contextual and Environmental Obstacles

These failures stem from the multimodal nature of lecture slides, where math exists alongside text, arrows, handwritten annotations, and diagrams.

- Multi-modality Issues: Non-mathematical elements often pollute the equation strings. In the Statistics course, a red arrow pointing to a variable was misinterpreted as an underbrace, while page numbers were frequently appended to the ends of equations like constants. Furthermore, phrases and annotations were not correctly transcribed using text{} blocks, causing letters to be rendered as a string of italicized variables without proper spacing.
- Boundary Errors: STEM lecture notes often derive formulas through subsequent, vertical steps. Models often struggle to define where one equation ends and another begins. In the Statistics course, two distinct equations were often merged into a single line, which could cause a screen reader to read two independent identities as a single continuous statement. The model also confuses figure annotations, signal labels, or physical descriptors with mathematical expressions. In the Electromagnetics course, a descriptor for an image was interpreted as part of an equation next to it.
- Excluded GTIs: Completely missed GTIs accounted for approximately 7.9% of the total ground truth instances. However, these were rarely random. They typically occurred when equations were embedded within tables, contained within diagrams, or partially covered by a large 'X' symbol drawn by the instructor.

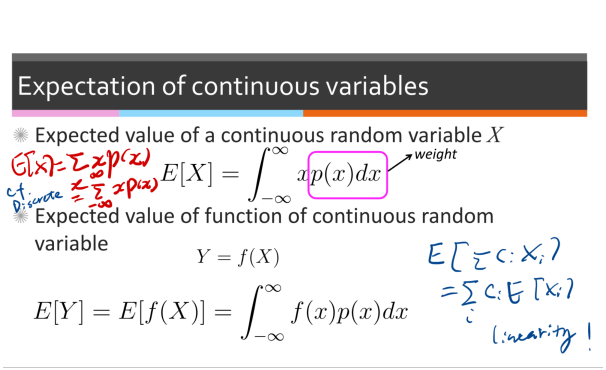


Figure 1: Example of a Lecture Slide with Environmental Obstacles and its output

(iii) Symbolic and Semantic Obstacles

This category involves 'Non-Recoverable' errors where the core identity of a symbol is lost, potentially altering the engineering context entirely.

- Ambiguous Character Substitution: Common substitutions involved infinity being interpreted as x, mu interpreted as m, u or a, and Gamma misinterpreted as rho. Greek letters and symbols that are not a part of the English language are particularly 'high risk' as they result in syntactically valid, but factually incorrect syntax. Given their prominence in mathematical notation in STEM content, this makes it a particularly difficult problem.
- Non-standard notation: Diagonal fractions and ||| (used to show logical flow) were frequently misinterpreted as ellipses (...) or stray lines, which destroyed the logical sequences in mathematical derivations.

Contextual Obstacles (Layout Accuracy Analysis)

Performance Metrics Aggregates

To assess the model's efficacy in detecting and preserving the visual organization of a slide, we use two primary metrics derived from our layout scoring rubric: Perfect Accuracy Rate and Complete Failure Rate. The Perfect Accuracy Rate (Score 3) requires the model to use the correct classification color and ensure no structural or environmental interference for the bounding box. Conversely, the Complete Failure Rate (Score 0) identifies instances where the model entirely fails to detect a region or produces an output that bears no relation to the ground-truth layout. These metrics allow us to distinguish between minor aesthetic overlaps (Scores 1 and 2) and significant failures in decomposing slides.

Course	Total Slides	Score 3	Score 2	Score 1	Complete Failure Rate	Perfect Accuracy Rate
Statistics	162	92	29	10	0.1914	0.5679
Information Systems	89	75	10	4	0	0.8427
Electromagnetics	42	21	14	7	0	0.5
Industrial Engineering	29	26	3	0	0	0.8966
Total	**322**	**214**	**56**	**21**	**0.0963**	**0.6646**

Table 5: Layout Performance Metrics Aggregates of MinerU-2.5 Across 4 STEM Courses

As illustrated in Table 5, MinerU-2.5 has significant variance in layout detection accuracy across the different courses. The model performed best in Industrial Engineering and Information Systems, achieving Perfect Accuracy Rates of 89.66% and 84.27% respectively, with 0% Complete Failure Rates in both. This success likely stems from these courses' slides having mostly structured text and typed equations, which present fewer layout ambiguities than handwritten annotations found in other disciplines.

In contrast, the Electromagnetics course presented the greatest challenge to layout fidelity, with the Perfect Accuracy Rate dropping to 50.00%. Furthermore, while the Statistics course had a higher raw number of perfect detections, it also had the highest Complete Failure Rate of 19.14%. This might be due to the increase in multimodal complexity. The model's layout-first approach also makes it susceptible to error propagation, where failures in initial region classification could have led to degraded performance in symbolic accuracy.

Taxonomy of Layout Obstacles

Based on the scoring of 322 slides, we identified three primary categories of layout failure that prevent a "Perfect" layout score.

(i) Classification and Type-Casting Errors

Classification accuracy is the most important component of layout detection, as it dictates the objective for the decoder in subsequent processing steps. Models sometimes prioritize broad document categories at the expense of nested instructional content.

- In-line Formulas: When mathematical expressions are embedded directly within standard text, the model often fails to recognize that specific region as math. Instead, the entire line is often classified as standard text, causing the system to parse symbols as standard alphanumeric characters.
- Listed Formula Exclusion: When content is organized into bulleted lists, the model classifies it using a dark green bounding box and frequently fails to identify nested math

within these lists. Formulas contained within diagrams or bulleted points are often ignored or excluded because the classification acts as a filter that prevents the HMER engine from scanning those regions for math.

(ii) Environmental Obstacles

These failures occur when the model's bounding boxes fail to isolate mathematical expressions from the multimodal background.

- Contextual Overlap: Bounding boxes often over-extend to include noise such as adjacent figure annotations or physical descriptors. This environmental interference causes the decoder to include unrelated descriptions and symbols in the same equation.
- Major Merges in High-Density Layouts: In courses with high multimodal complexity, such as Statistics, the model frequently combined three or more distinct ground-truth instances into a single box. This breaks the intended reading order and merges independent logical steps or derivations into a single string.

Completely Omitted Slides

While MinerU-2.5 demonstrated high accuracy for standard symbolic notation, a substantial percentage of the lecture slides were not converted to LaTeX. To identify the underlying obstacles and types of Math involved, we analyzed these slides by comparing MinerU-2.5's layout output against ground-truth LaTeX code the instructor would manually write for the equations. This comparison revealed the following major issues that could potentially have caused the slides to be omitted:

1. The layout couldn't identify the equation/s as Math (see: Appendix C, Figure C1)
2. Typed equation requires nested parentheses
3. The handwritten equations might be hard to transcribe, even if they are simple, ie. (see: Appendix C, Figure C2)
4. The handwritten equation has a two-dimensional display such as a fraction, choose, overline, hat, or multiple equation cases
5. Handwritten equations have matrices
6. Handwritten equations are mixed with text, annotations, arrows, call outs, cdot, rvert
7. Handwritten equations are adjacent to graphs or text

Further investigation into the types of math involved in these equations showed that many of these handwritten equations involve LaTeX language structures that are considered advanced. For example, in the LaTeX wikibook [14], text mixed with equations is considered 'Advanced Math' and is accommodated by the AMS LaTeX package "amsmath". The 'Advanced Math' section from this package consists of the following major categories of structures:

1. Equation numbering
2. Vertically aligning displayed mathematics
3. Indented equations
4. Boxed equations
5. Custom operators
6. Advanced formatting
7. Text in aligned math display

Based on the results from the existing software, we think these advanced LaTeX structures have not yet been employed in the conversion of handwritten equations. In our lecture slides dataset, two of these structures are prevalent: the vertically aligning displayed math and the text in aligned math display (see: Appendix C, Figures C3 and C4). Note that the LaTeX code generated by the instructor was done without using the amsmath package.

Suggestions and Guidelines

Guidelines for Educators when creating Learning content

To ensure machine-readable transcriptions, instructors should focus on reducing spatial arrangement inaccuracies in the converted equations. This can be achieved by treating the slide layout as a structured document rather than spontaneously adding annotations.

1. Extra Whitespace as a Delimiter: Use generous whitespace between independent mathematical statements and words. When the steps are written too closely, models often ”collapse” them into a single unreadable string or misinterpret the boundaries.
2. Conflict-Free Annotations: Avoid drawing arrows, circles, or cross-outs that physically overlap with equations. These environmental interferences often cause the model to hallucinate symbols.
3. Standardized Vertical Divisions: For derivations, use a clear vertical alignment for each step. If possible, use horizontal lines to separate distinct sections to prevent ”boundary errors” where multiple equations are merged into one.
4. High-Contrast Notation: Distinguish clearly between ”high-risk” characters, such as v versus nu. Ensuring that summation and integral bounds are written in a slightly smaller, distinct vertical position could help the model identify them as limits rather than standard subscripts.

Outlining challenging examples for developers and researchers to create/improve domain-specific testing datasets

The goal for developers working in HMER research should be to move beyond raw symbol detection and focus on modeling the 2D structural relationships inherent in STEM content.

1. Ambiguous Pairs Testing: Training datasets should be enriched with pairs that are frequently substituted, particularly Greek letters that mimic Latin counterparts.
2. Support for ”Text-within-Math”: Models currently struggle with phrases embedded in equations. Developing models that automatically apply blocks for non-symbolic annotations would prevent the italicized variable strings that break screen-reader interpretability.
3. 2D Layout Benchmarking: Shift focus toward benchmarking ”high-risk” 2D syntax, such as nested fractions, large matrices, and multiline alignment environments like amsmath. These structures are where current VLMs typically experience ”spatial failures”.

Conclusion

Our analysis shows that while MinerU-2.5 is generally effective at recovering the mathematical meaning of instructional equations across diverse STEM lecture slides, its performance degrades when the mathematical content relies on spatial organization, multimodal complexity, or advanced LaTeX structures. Across 369 ground-truth instances drawn from statistics, information systems, industrial engineering, and electromagnetics courses, the 14.6% gap between Strict and Lenient Instance Rates reveals a recurring pattern: equations are often symbolically correct but structurally degraded. This indicates that recognition errors are more often tied to layout and presentation than to symbol detection alone.

Furthermore, our layout analysis across 322 slides underscores how failures in page-level decomposition can lead to symbolic errors. With an overall 66.5% Perfect Accuracy Rate, the model frequently struggles with classification and type casting, such as misidentifying in-line formulas as standard text or flagging complex handwritten matrices as images. These layout-level obstacles, including a 9.6% Complete Failure Rate in high-density slides, show that the model’s ”layout-first” approach is susceptible to error propagation when working with multimodal boundaries.

Structural and spatial obstacles — such as collapsed spacing, misinterpreted subscripts and superscripts, and dropped summation or integral bounds — directly affect readability and accessibility, particularly for screen readers. Contextual interference from arrows, annotations, diagrams, and surrounding text further complicates recognition, leading to boundary errors in which equations are merged or contaminated by non-mathematical elements. Excluded equations, comprising roughly 7.9% of instances, are not random but are strongly associated with handwritten, two-dimensional, or layout-complex mathematics, especially when equations are embedded in tables, diagrams, or aligned multi-line derivations.

Overall, the results indicate that current HMER/VLM systems are bottlenecked less by symbol recognition and more by their limited ability to model spatial hierarchy, multimodal boundaries,

and advanced LaTeX constructs such as aligned displays and text-within-math. Thus, future work should emphasize evaluation and model design that explicitly separates symbolic correctness from structural representation.

Acknowledgment

This work was supported in part by a GIANT 2025-26 grant sponsored by the Institute for Inclusion, Diversity, Equity and Access (IDEA) in the Grainger College of Engineering at the University of Illinois Urbana-Champaign. Additional support was provided by Disability Resources and Educational Services (DRES) at the University of Illinois Urbana-Champaign. The human subject study was permitted by #IRB25-1123 at the University of Illinois Urbana-Champaign.

References

[1] L. Asanaka *et al.*, "Improving the Accessibility of Mathematical and other STEM Content in Engineering courses through Machine Learning Models," in *ASEE Annual Conference & Exposition*, 2025.

[2] X. Ding *et al.*, "Understanding the needs of students to make Mathematics and other STEM Contents more accessible in college Engineering courses," in *ASEE Annual Conference & Exposition*, 2025.

[3] X. Ding *et al.*, "Evaluating the low-stakes assessment performance, student perceived accessibility, belongingness, and self-efficacy in connection to the use of digital notes in engineering and computing courses," in *ASEE Annual Conference & Exposition*, 2023.

[4] J. Niu *et al.*, "MinerU2.5: A Decoupled Vision-Language Model for Efficient High-Resolution Document Parsing,", 2025. [Online]. Available: https://arxiv.org/abs/2509.22186

[5] J. Spracklen *et al.*, "We Have a Package for You! A Comprehensive Analysis of Package Hallucinations by Code Generating LLMs," in *USENIX Security Symposium*, 2025. [Online]. Available: https://www.usenix.org/system/files/usenixsecurity25-spracklen.pdf

[6] Arxiv.org, "MathBridge: A Large-Scale Dataset for Translating Mathematical Expressions into Formula Images,", 2023. [Online]. Available: https://arxiv.org/html/2408.07081v1

[7] D. Matsumoto and T. Nakai, "Syntactic theory of mathematical expressions," *Cognitive Psychology*, vol. 146, p. 101606, Sep. 2023.

[8] Z. Ye, "An Intelligent-Detection Network for Handwritten Mathematical Expression Recognition,", 2023. [Online]. Available: https://arxiv.org/abs/2311.15273

[9] A. Fadeeva, V. Coriou, D. Antognini, C. Musat, and A. Maksai, "InkFM: A Foundational Model for Full-Page Online Handwritten Note Understanding,", 2025. [Online]. Available: https://arxiv.org/abs/2503.23081

[10] Y. Xie *et al.*, “ICDAR 2023 CROHME: Competition on Recognition of Handwritten Mathematical Expressions,” in *Lecture Notes in Computer Science (ICDAR 2023)*, 2023, pp. 553–565.

[11] Arxiv.org, “The Return of Structural Handwritten Mathematical Expression Recognition,”, 2023. [Online]. Available: https://arxiv.org/html/2508.19773v1

[12] H. Li, J. Li, J. Cao, Z. Yang, and Y. Xiong, “Towards Scalable Training for Handwritten Mathematical Expression Recognition,”, 2025. [Online]. Available: https://arxiv.org/html/2508.09220v1

[13] C. Cui *et al.*, “PaddleOCR-VL: Boosting Multilingual Document Parsing via a 0.9B Ultra-Compact Vision-Language Model,”, 2025. [Online]. Available: https://arxiv.org/abs/2510.14528

[14] Wikimedia Contributors, “LaTeX/Mathematics,” Wikibooks.org, Sep. 2005. [Online]. Available: https://en.wikibooks.org/wiki/LaTeX/Mathematics

Appendix

Appendix A — State of Art of HMER Models

Current HMER models can be broadly classified into four distinct architectural categories, each offering varying levels of efficacy for handwritten STEM content:

(i) Specialized Pipeline Systems

Traditional pipeline systems, such as MinerU [A1] and Docling [A2], initially rely on a layout detector to isolate mathematical regions before passing them to an OCR engine, such as PaddleOCR, for LaTeX conversion. While these systems excel at detecting the location of a formula on a page, they are highly prone to 'error propagation', where a missed bounding box results in the complete omission of mathematical content.

(ii) Neural Encoder-Decoder Models

Models like UniMERNet [A3], Pic2Tex, and TexTeller [A4] utilize a Vision Transformer (ViT) encoder to detect symbolic patterns and a Transformer decoder to generate LaTeX tokens. However, architectural variations, like using different modules, or training processes can make neural models ideal for different tasks. For instance, UniMERNet introduces character-level awareness through a Fine-Grained Embedding (FGE) module, which prevents the model from arbitrarily "chopping" the image and preserves characters within the feature map, improving spatial relationships. Conversely, TexTeller applies "scaling laws" to math recognition, which achieves high precision on clean, symbolic grammar, but often underperforms on noisy, non-standard scribbles characteristic of handwritten lecture annotations.

(iii) Self-Supervised Models and Stroke-based Models

To address the scarcity of labeled handwritten data, models like Art4Math [A5] and MoCoMER [A6] utilize self-supervised pre-training. Art4Math, for instance, pre-trains on widely-available human sketches and leverages the fact that both handwritten math and sketches are sparse, symbolic abstractions rendered with strokes [A5]. Similarly, InkFM utilizes 'temporal data' or stroke order for scrawl [A7], which is highly effective for tablet-based handwriting, but less optimized for static, scanner slide archives.

(iv) Vision-Language Models (VLMs)

Modern VLMs, such as Qwen2.5-VL-72B [A8] and the MinerU-2.5 [A9], have shifted toward 'vision-first' processing. Unlike neural encoder-decoder models, these VLMs use semantic context to resolve visual ambiguity. For instance, if a surrounding sentence mentions 'velocity', the VLM can more accurately distinguish a handwritten v from a nu. However, massive models like Qwen2.5-VL-72B often struggle with strict output formatting requirements and multiline formulas.

References

[A1] B. Wang *et al.*, "MinerU: An Open-Source Solution for Precise Document Content Extraction,", 2024. [Online]. Available: https://arxiv.org/abs/2409.18839

[A2] IBM Research, "Docling Technical Report,",2022. [Online]. Available: https://arxiv.org/html/2408.09869v1

[A3] B. Wang *et al.*, “UniMERNet: A Universal Network for Real-World Mathematical Expression Recognition,”, 2024. [Online]. Available: https://arxiv.org/abs/2404.15254

[A4] H. Li, J. Li, J. Cao, Z. Yang, and Y. Xiong, “Towards Scalable Training for Handwritten Mathematical Expression Recognition,”, 2025. [Online]. Available: https://arxiv.org/html/2508.09220v1

[A5] Y. Zhou *et al.*, “Art4Math: Handwritten Mathematical Expression Recognition via Multimodal Sketch Grounding,” in *Proc. 33rd ACM Int. Conf. Multimedia*, Oct. 2025, pp. 1549–1558.

[A6] S. Pramanik, S. Mitra, and N. Das, “MoCoMER: A Self Supervised Representation Learning for Handwritten Mathematical Expression Recognition,” in *Proc. Fifteenth Indian Conf. Computer Vision Graphics and Image Processing*, Dec. 2024, pp. 1–10.

[A7] A. Fadeeva, V. Coriou, D. Antognini, C. Musat, and A. Maksai, “InkFM: A Foundational Model for Full-Page Online Handwritten Note Understanding,”, 2025. [Online]. Available: https://arxiv.org/abs/2503.23081

[A8] S. Bai *et al.*, “Qwen2.5-VL Technical Report,”, 2025. [Online]. Available: https://arxiv.org/abs/2502.13923

[A9] J. Niu *et al.*, “MinerU2.5: A Decoupled Vision-Language Model for Efficient High-Resolution Document Parsing,”, 2025. [Online]. Available: https://arxiv.org/abs/2509.22186

Appendix B — Model Selection

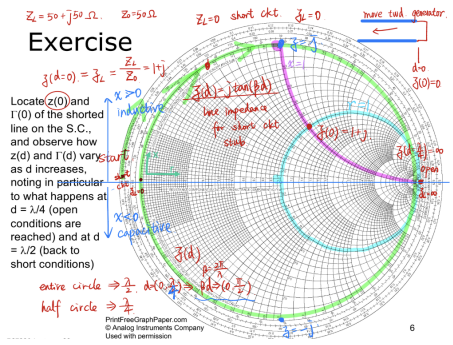
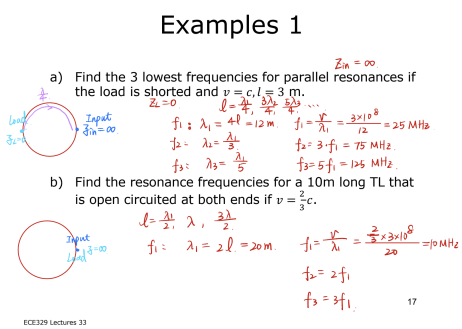
<table>
<tr><th>Index</th><th>Lecture Slides vs OCR Outputs</th></tr>
<tr><td rowspan="3">1</td><td> 
</td></tr>
<tr><td>MinerU-2.5 Output</td><td>PaddleOCR-VL Output</td></tr>
<tr><td>Exercise
$z(d=0)=z_2=\frac{z_2}{z_0}=1+j$
Locate $z\,(0)$ and $\Gamma\,(0)$ of the shorted line on the S.C., and observe how $z\,(d)$ and $\Gamma\,(d)$ vary as d increases, noting in particular to what happens at $d=\lambda/4$ (open conditions are reached) and at $d=\lambda/2$ (back to short conditions)
entirecircle $\Rightarrow \frac{\lambda}{2}, d=(0,\frac{\lambda}{4}) \Rightarrow \beta d \Rightarrow (0,\frac{\pi}{2})$
halfcircle $\Rightarrow \frac{\lambda}{4}$.
PrintFreeGraphPaper.com
⊙ Analog Instruments Company
Used with permission
$x>0$</td><td>Exercise
$z(d=0)=z_L=\frac{z_L}{z_0}=1+j$
Locate $z(0)$ and $\Gamma(0)$ of the shorted line on the S.C., and observe how $z(d)$ and $\Gamma(d)$ vary as d increases, noting in particular what happens at $d=\lambda/4$ (open conditions are reached) and at $d=\lambda/2$ (back to short conditions)
half circle $\Rightarrow \frac{\lambda}{4}$
PrintFreeGraphPaper.com © Analog Instruments Company</td></tr>
<tr><td rowspan="3">2</td><td> 
</td></tr>
<tr><td>MinerU-2.5 Output</td><td>PaddleOCR-VL Output</td></tr>
<tr><td>Examples 1
$Z_{in}=\infty$.
(a) Find the 3 lowest frequencies for parallel resonances if the load is shorted and $v=c, l=3$ m.
Input
$f_{in}=\infty$
$z_L=0. \quad l=\frac{\lambda_1}{4},\frac{3\lambda_2}{4},\frac{5\lambda_3}{4},\ldots$
$f_1:\lambda_1=4l=12m. f_1=\frac{v}{\lambda_1}=\frac{3\times10^8}{12}=25MHz$
$f_2=\lambda_2=\frac{\lambda_1}{3}.$
$f_3:\quad \lambda_3=\frac{\lambda_1}{5}$
$f_2=3\cdot f_1=75MHz.$
$f_3=5f_1=125\text{MHz}.$
b) Find the resonance frequencies for a 10 m long TL that is open circuited at both ends if $v=\frac{2}{3}c$.
$l=\frac{\lambda_1}{2},\lambda,\frac{3\lambda}{2}$
$f_1:\quad \lambda_1=2l.=20m. \quad f_1=\frac{v}{\lambda_1}=\frac{\frac{2}{3}\times3\times10^8}{20}=10MHz$
$f_2=2f_1$
$f_3=3f_1.$</td><td>Examples 1
$z_{in}=\infty$.
a) Find the 3 lowest frequencies for parallel resonances if the load is shorted and v = c, l = 3 m.
Input
$f_{in}=\infty$
$z_L=0 \quad l=\frac{\lambda_1}{4},\frac{3\lambda_2}{4},\frac{5\lambda_3}{4},\ldots$
$f_1:\lambda_1=4l=12m. \quad f_1=\frac{v}{\lambda_1}=\frac{3\times10^8}{12}=25MHz$
$f_2=\lambda_2=\frac{\lambda_1}{3} \quad f_2=3\cdot f_1=75MHz.$
$f_3:\lambda_3=\frac{\lambda_1}{5} \quad f_3=5f_1=125MHz.$
b) Find the resonance frequencies for a 10m long TL that is open circuited at both ends if $v=\frac{2}{3}c$.
$l=\frac{\lambda_1}{2},\lambda,\frac{3\lambda}{2}, f_1=\lambda_1=2l=20m, \quad f_1=\frac{v}{\lambda_1}=\frac{\frac{2}{3}\times3\times10^8}{20}=10, f_2=2f_1, f_3=3f_1.$</td></tr>
</table>

Table B1: MinerU-2.5 v/s PaddleOCR-VL Markdown Rendering

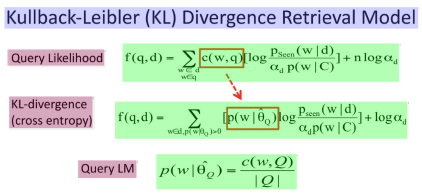
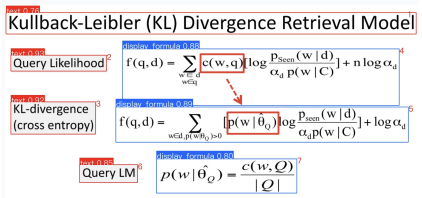
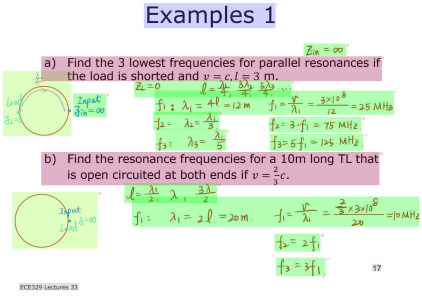
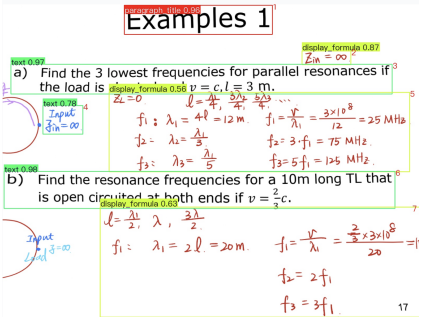
<table>
<tr><th>Index</th><th>Lecture Slides vs OCR Outputs</th></tr>
<tr><td>1</td><td>
<h2>Kullback-Leibler (KL) Divergence Retrieval Model</h2>
Query Likelihood $f(q,d) = \sum_{w \in d, w \in q} c(w,q)[\log \frac{p_{Seen}(w|d)}{\alpha_d p(w|C)}] + n \log \alpha_d$
KL-divergence (cross entropy) $f(q,d) = \sum_{w \in d, p(w|\theta_Q)>0} [p(w|\hat{\theta}_Q)][\log \frac{p_{seen}(w|d)}{\alpha_d p(w|C)}] + \log \alpha_d$
Query LM $p(w|\hat{\theta}_Q) = \frac{c(w,Q)}{|Q|}$
<table>
<tr><th>MinerU-2.5 Output</th><th>PaddleOCR-VL Output</th></tr>
<tr><td>  </td><td>  </td></tr>
</table>
</td></tr>
<tr><td>2</td><td>
<h2>Examples 1</h2>
a) Find the 3 lowest frequencies for parallel resonances if the load is shorted and $v = c, l = 3$ m.
b) Find the resonance frequencies for a 10m long TL that is open circuited at both ends if $v = \frac{2}{3}c$.
ECE329 Lectures 33
<table>
<tr><th>MinerU-2.5 Output</th><th>PaddleOCR-VL Output</th></tr>
<tr><td>  </td><td>  </td></tr>
</table>
</td></tr>
</table>

Table B2: MinerU-2.5 v/s PaddleOCR-VL Layout Detection

Appendix C — Completely Omitted Slide Examples

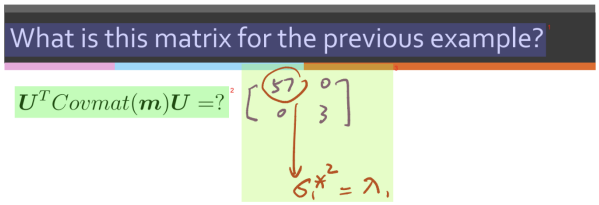


Figure C1: The matrix in yellow-green was identified as an image.

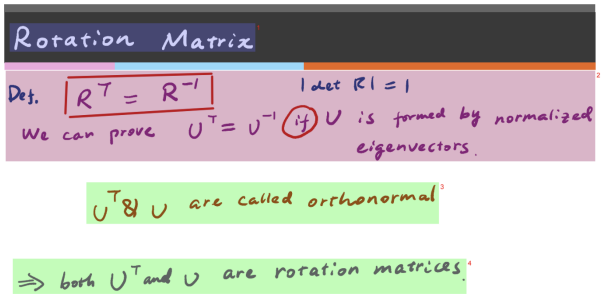
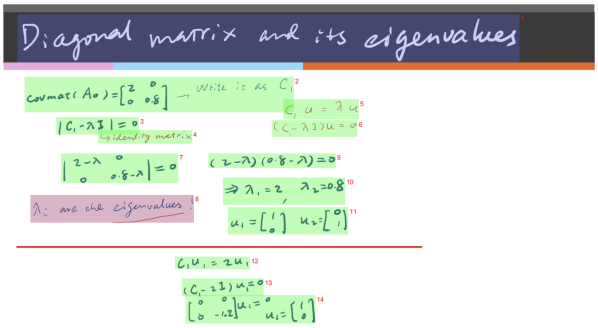


Figure C2: Handwritten equations were identified as text, the equations are relatively easy.



Equation/Expression	LaTeX Code
$covmat(A_0) = \begin{bmatrix} 2 & 0 \\ 0 & 0.8 \end{bmatrix} \longrightarrow$ write it as C_1	`covmat(A_0) = \begin{bmatrix} 2 & 0 \\ 0 & 0.8 \end{bmatrix} \longrightarrow \text{write it as}\ C_1`
$\|C_1 - \lambda I\| = 0$ $\begin{vmatrix} 2-\lambda & 0 \\ 0 & 0.8-\lambda \end{vmatrix} = 0$ $(2-\lambda)(0.8-\lambda) = 0 \Rightarrow \lambda_1 = 2, \lambda_2 = 0.8$	`|C_1 - \lambda I| = 0 \\ \begin{vmatrix} 2-\lambda & 0 \\ 0 & 0.8-\lambda \end{vmatrix} = 0 \\ (2-\lambda)(0.8-\lambda) = 0 \Rightarrow \lambda_1 = 2, \lambda_2 = 0.8`
λ_i are the eigenvalues	`\lambda_i \quad \text{are the eigenvalues}`

Figure C3: The layout of the bounding boxes of a handwritten slide and part of the LaTeX code (These equations were not converted at all although the bounding boxes are correct).

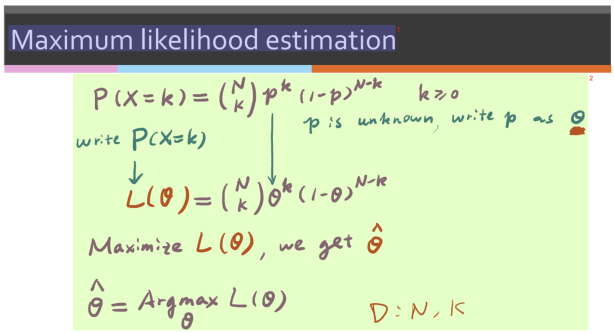


Figure C4: The text and equations on the slide were classified as an image, resulting in no converted equations.