TexOCR: Advancing Document OCR Models for Compilable Page-to-LaTeX Reconstruction

Chengye Wang [Z] **Lin Fu** [Z] **Zexi Kuang** [Y] **Yilun Zhao** [Y]

[Y] Yale University [Z] Zhejiang University

Data & Models Code

Abstract

Existing document OCR largely targets plain text or Markdown, discarding the structural and executable properties that make LaTeX essential for scientific publishing. We study page-level reconstruction of scientific PDFs into compilable LaTeX and introduce TexOCR-Bench, a benchmark, and TexOCR-Train, a large-scale training corpus, for this task. TexOCR-Bench features a multi-dimensional evaluation suite that jointly assesses transcription fidelity, structural faithfulness, and end-to-end compilability. Leveraging TexOCR-Train, we train a 2B-parameter model, TexOCR, using supervised fine-tuning (SFT) and reinforcement learning (RL) with verifiable rewards derived from LaTeX unit tests that directly enforce compilability and referential integrity. Experiments across 21 frontier models on TexOCR-Bench show that existing systems frequently violate key document invariants, including consistent section structure, correct float placement, and valid label–reference links, which undermines compilation reliability and downstream usability. Our analysis further reveals that RL with verifiable rewards yields consistent improvements over SFT alone, particularly on structural and compilation metrics.

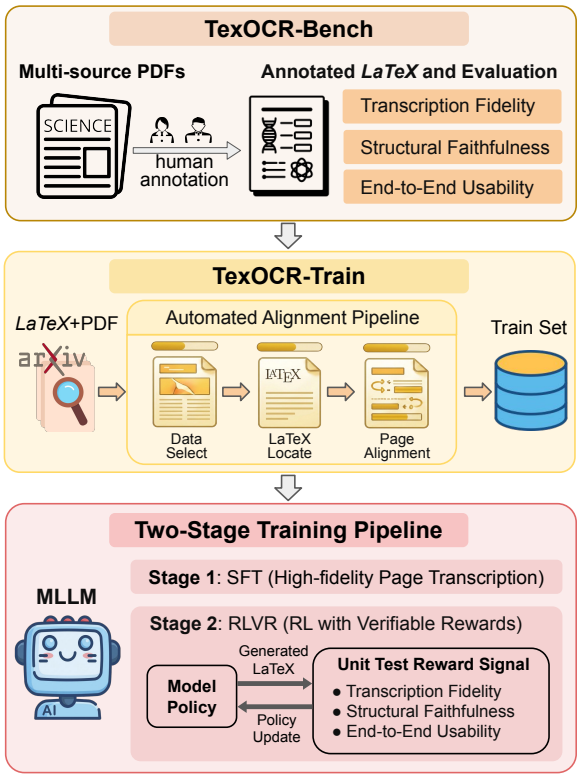

Figure 1: Overview of our work, covering data synthesis, model training, and multi-dimensional evaluation.

1 Introduction

Scientific knowledge remains predominantly distributed as PDF, yet most downstream workflows depend on editable, structure-preserving representations that support retrieval, reuse, and reproducible publishing (for example, compilation-ready LaTeX) (Lo et al., 2020). Traditional OCR systems have historically been modular, combining document analysis, layout parsing, text recognition, and post-processing rules (PDF Association staff, 2015; Wick et al., 2018; Saha et al., 2010). More recently, OCR has shifted toward multimodal large language models (MLLMs) that can transcribe directly from document images (Greif et al., 2025; Inoue, 2025; Chen et al., 2025b; Wang et al., 2025; Chen et al., 2025a), raising the ceiling on end-to-end recognition quality and enabling broader general-purpose and OCR-specialized systems.

Despite these advances, reliable page-level reconstruction of scientific papers into executable LaTeX remains under-served. Existing work primarily linearizes PDFs into plain text or Markdown (Li et al., 2025b; Heakl et al., 2025). When LaTeX is considered, it is typically framed as a localized conversion task for specific regions, most commonly math equations and tables (Poznanski et al., 2025a; Jiang et al., 2025; Ling et al., 2025). These approaches under-specify the global constraints that determine whether a reconstructed project is actu-

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 35821–35845

ally usable: cross-page structure, stable float placement, consistent numbering, and reference integrity across citations, equations, figures, tables, and sections. As a result, even when surface transcription quality appears high, reconstructed LaTeX sources can remain brittle. Minor errors that are inconsequential in plain text, such as a missing brace, an unescaped character, an incorrect environment boundary, or a mismatched label, can render the document uncompilable or silently corrupt semantics by redirecting references and numbering.

To bridge this gap, we introduce TEXOCR-Bench, a new evaluation benchmark for end-to-end, page-level LaTeX reconstruction with an explicit emphasis on transcription accuracy, executability, and structural fidelity. TEXOCR-Bench contains 2,135 expert-annotated examples spanning diverse document types and technical domains. To move beyond surface transcription scores, we design a comprehensive evaluation protocol that jointly measures transcription fidelity, structural faithfulness, and end-to-end usability. The protocol includes component-level checks (*e.g.,* text, table, and formula accuracy), structure-aware checks (*e.g.,* section accuracy, citation coverage, and figure/table reference validity), and a compilation success rate computed by compiling the reconstructed project without manual intervention.

To facilitate future research, we release a large-scale training dataset, TEXOCR-Train, and a training recipe tailored to compilable LaTeX reconstruction rather than generic OCR. We collect and process arXiv LaTeX source archives paired with PDFs at scale and construct page-aligned supervision, resulting in 57K papers and 404K (page image, LaTeX) training pairs. Building on this corpus, we train TEXOCR on Qwen3-VL-2B (Team, 2025) with SFT for high-fidelity page transcription, and then further optimize it using reinforcement learning with verifiable rewards (RLVR). To adopt reliable reward functions for effective LaTeX-focused model development, we design binary unit tests that mirror the nine evaluation metrics in TEXOCR-Bench, spanning transcription fidelity, structural faithfulness, and end-to-end usability.

We conduct a systematic evaluation of 21 frontier MLLMs and OCR models on TEXOCR-Bench. Our results show that TEXOCR-Bench poses a substantial challenge: even the best-performing model, GPT-5.3, achieves only 78.5 accuracy. While OCR systems such as olmOCR2 (Poznanski et al., 2025b) and DeepSeek-OCR (Wei et al., 2025) perform competitively on previous PDF-to-Markdown benchmarks, they degrade markedly on TEXOCR-Bench, indicating that reconstructing compilable, structure-consistent LaTeX requires capabilities beyond high-quality transcription. Among open-source models, TEXOCR attains state-of-the-art performance on TEXOCR-Bench. Moreover, we find that our RLVR scheme yields consistent improvements over SFT across both component-level and structure-aware metrics, driven by verifiable unit-test rewards that directly target compilation and referential integrity. Qualitative analysis further suggests that RLVR encourages the model to internalize LaTeX "invariants" (*e.g.,* balanced delimiters, well-formed environments, and stable label–reference alignment), reducing brittle failure modes that otherwise prevent successful compilation or silently corrupt document semantics.

Our main contributions are summarized below:

- We introduce TEXOCR-Bench, a new benchmark for end-to-end, page-level reconstruction of scientific PDFs into compilable LaTeX, with 2,135 expert-annotated examples spanning diverse document types and technical domains.
- We propose a comprehensive evaluation protocol that goes beyond surface similarity by jointly measuring component-level transcription fidelity, structure-aware consistency, and end-to-end usability via zero-touch compilation success.
- We release TEXOCR-Train, a large-scale training corpus, along with a training recipe that combines SFT and RLVR methods. Using this recipe, we train a 2B model, TEXOCR.
- We conduct a systematic evaluation of 21 frontier models on TEXOCR-Bench, showing that TEXOCR achieves the strongest performance among open-source baselines and that our RLVR design delivers consistent improvements on both component- and structure-level metrics.

2 Related Works

OCR Evaluation Benchmarks. Recent benchmarks evaluate document OCR along two complementary axes. (1) *End-to-end conversion from PDF to Markdown or plain text*: READoc (Li et al., 2025b) and OmniDocBench (Ouyang et al., 2025) define unified protocols for realistic multi-page extraction across diverse PDFs, while olmOCR-Bench (Poznanski et al., 2025b) adds pass or fail unit tests that check key document properties such

as text presence, reading order, tables, formulas, and baseline functionality. (2) *Structured element transcription to LaTeX for tables and formulas*: Table2LaTeX-RL (Ling et al., 2025) evaluates table reconstruction with layout-consistent scoring and render-based checks, and CMER-Bench (Bai et al., 2025b) benchmarks formula transcription with difficulty stratification. Across these settings, evaluation typically combines string or token similarity, consistency checks, and unit-test pass rates. In contrast, we introduce a new OCR task and a more comprehensive protocol that complements component metrics with structural checks and a zero-touch compilation test, yielding a stricter measure of end-to-end usability.

Document OCR Methods. Document OCR has increasingly shifted from modular pipelines to MLLMs that transcribe directly from document images. Early studies showed that encoder–decoder MLLMs can convert scientific PDFs into plain text without explicit layout supervision (Blecher et al., 2023; Wei et al., 2024). Recent approaches include end-to-end generation of linearized text or markup (Nassar et al., 2025; Liu et al., 2025; Li et al., 2025a; Duan et al., 2026; Dong et al., 2026; Wu et al., 2026; Zheng et al., 2026; Taghadouini et al., 2026) as well as hybrid systems that use MLLMs as recognition backbones within conventional document processing frameworks (Wei et al., 2025; Cui et al., 2025; Wang et al., 2026; Wei et al., 2026). Earlier systems mainly relied on large-scale SFT on paired document images and text, such as Nanonets-OCR2 (Mandal et al., 2025) and olmOCR (Poznanski et al., 2025a), whereas newer work increasingly adopts RL with task-specific or verifiable rewards to improve consistency on tables, formulas, and reading order like DianJin-OCR-R1 (Chen et al., 2025a), olmOCR2 (Poznanski et al., 2025b). In contrast, we target compilable page-to-LaTeX reconstruction and introduce benchmarking and training signals that directly optimize executability and global LaTeX invariants beyond surface transcription.

3 TEXOCR-Bench Benchmark

This section first discusses the evaluation suite of TEXOCR-Bench, and then details the benchmark construction process.

3.1 Evaluation Suite

To comprehensively assess both OCR quality and LaTeX reconstruction fidelity, we organize nine metrics along three complementary dimensions:

Transcription Fidelity. It measures fine-grained content recovery from the page image. **Complex Text Preservation (CTP)** extracts one plain-text sentence per section (excluding LaTeX-specific commands) from the reference source and checks whether each sentence appears in the generated output, with strict matching on case sensitivity, punctuation, and character-level accuracy. **Formula Accuracy (FA)** removes labels, environments, and layout-related noise from mathematical expressions, applies expression normalization, and compares formulas via string-based sequence matching. **Table Accuracy (TA)** extracts and normalizes numerical entries in tables, then matches generated tables against the ground truth based on (i) the overlap ratio of numerical entries and (ii) the hit rate of *unique numbers*.

Structural Faithfulness. It checks document-level structure and cross-reference consistency. Specifically, **Section Accuracy (SA)** verifies whether all section titles are correctly recovered and aligned with the original document structure, including both title text matching and hierarchical correctness (*e.g.,* proper restoration of `\section`). **Citation Coverage (CC)** extracts bibliography information from the aggregated LaTeX and evaluates whether each in-text citation is correctly generated, checking the correctness of `\cite{}` keys, their positions in the text, and the presence of missing or misrecognized citations. **Reference Validity (RV)** checks whether figure and table references conform to valid LaTeX syntax and semantics, detecting undefined references and verifying that commands like `\ref{fig:figure_x}` correctly resolve to existing labels.

End-to-End Usability. It evaluates overall output quality and executability. **Document-Level Similarity (DS)** concatenates the original LaTeX source into a single reference file and computes the normalized character-level edit distance between the generated LaTeX and the ground truth as $\text{Similarity} = 1 - \text{Levenshtein}(A, B)/\max(|A|, |B|)$, where A and B denote the reference and generated texts. **Baseline** identifies severe page-level generation fail-

Set	#Documents	#Figures (Conv.)	#Images	#Tables	#Formulas	Average LaTeX Length
TEXOCR-Train	57K	404K	181K	231K	488K	31.8K
TEXOCR-Bench	2K	14.5K	7.5K	13.9K	31K	39.1K

Table 1: Dataset statistics for TEXOCR-Train and TEXOCR-Bench. #Figures (Conv.) denotes figures converted from PDF pages. LaTeX Length reports the average length of the corresponding LaTeX source.

ures through basic usability checks, such as large-scale truncation, token loss, or abnormal characters. **Compilation Success Rate (CSR)** integrates all generated LaTeX fragments into a complete project and compiles it using a standard LaTeX pipeline without manual intervention, reflecting overall usability, engineering completeness, and practical readiness of the output.

3.2 Sourcing Documents and Creating Tests

Given the evaluation suite above, we construct TEXOCR-Bench in two stages: sourcing a diverse pool of PDF documents, and expert annotation followed by test-set assembly.

Data Sources. We collect source documents in PDF format from multiple heterogeneous sources, covering both contemporary scientific articles and scanned materials (Poznanski et al., 2025a). Specifically, the data sources include: (1) **arXiv** from which we collect scientific papers with publicly available PDF sources; (2) **Public-domain mathematics textbooks**, where we randomly sample pages from each document to capture diverse mathematical notation and formatting styles; (3) **The Library of Congress digital archives**, from which we sample historical letters and typewritten documents that come with existing human transcriptions; (4) **The Internet Archive**, which serves as a large repository of public-domain scanned documents, including journals, government reports, and technical manuals.

Document Selection and Annotation. From this pool, we select 2,135 documents of moderate length with rich LaTeX-relevant structures, ensuring comprehensive coverage of the evaluation dimensions introduced above, such as figures, tables, mathematical formulas, and hierarchical section organizations. These selected documents constitute the initial dataset. Given the selected PDFs, we assign annotation tasks to human annotators, who are instructed to transcribe the corresponding content into LaTeX format following the guidelines detailed in Appendix B. We then construct the test set by consolidating each annotated LaTeX project into a single `.tex` file using rule-based heuristics. From the merged source, we extract figures, tables, equations, and section headings with lightweight parsers, which serve as ground-truth annotations for metric computation. Detailed implementations of all evaluation metrics are provided in Appendix A.

4 TEXOCR-Train Construction

To enable training at scale, we release TEXOCR-Train. The following subsections detail the automated data construction process.

4.1 Paper Collection

To construct our benchmark, we programmatically collect official `.tar.gz` LaTeX sources and PDFs from arXiv, covering papers submitted between January 2022 and October 2025. For each paper, we resolve cross-file dependencies and merge all reachable `.tex` files into a single canonical source while preserving document order and file provenance. On this unified representation, we apply a deterministic rule-based parser to recover document structure, including section hierarchy as well as figure and table constructs. These normalized structural annotations provide standardized inputs and supervision for downstream benchmark tasks.

4.2 Train Set Design

To construct a high-quality training dataset for PDF-to-LaTeX conversion, we design a structured alignment and labeling pipeline based on the collected corpus. Each document is segmented into a set of single-page screenshots and training samples are organized as paired instances of (page image, textual supervision).

Page–Body Alignment. We use reference markers to find where the references start. Everything before that is treated as the main body, and only the first reference page is kept as the reference section. For pages in the main body, we use GPT5-mini (OpenAI, 2025) to detect begin/end token boundaries of the page content and align them to the crawled LaTeX source.

Float Placement. A key challenge is that the rendered PDF layout does not always align with the linear order of the LaTeX source: figures and tables defined within a given section may appear on different pages due to float placement or pagination. To resolve this mismatch, we use pdf2figure (Clark and Divvala, 2016) to detect and localize figures and tables across the entire PDF, and leverage the resulting global layout information to assign each element to its most appropriate page, yielding a consistent mapping between each page image and its complete LaTeX representation.

Bibliography Supervision. For the reference section, our supervision target is not the LaTeX-rendered bibliography list but rather BibTeX entries. Accordingly, we use GPT-based OCR outputs as labels for reference pages. For the first page containing references, we apply a hybrid labeling strategy: non-reference regions are supervised with standard LaTeX, whereas reference regions are converted into the corresponding BibTeX format. This design encourages the model to learn a structured switch from body LaTeX generation to bibliography entry generation near the end of the document. Then we apply this pipeline to all papers, resulting in 57K papers and 404K training pairs (page image, LaTeX/BibTeX text).

5 Two-Stage Training Pipeline

We train our PDF-to-LaTeX model in two stages: SFT for faithful page-level transcription, followed by RLVR to target failure modes such as formula rendering and table structure that are poorly captured by token-level likelihood.

5.1 Stage I: SFT

We initialize from a strong instruction-tuned multimodal backbone and fine-tune it on our page-aligned pairs (x, y), where x is a single-page PDF screenshot and y is the corresponding LaTeX (or BibTeX for reference regions). Given an instruction prompt p (describing the desired output format and conventions), the model is trained with standard next-token prediction:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\sum_{t=1}^{|y|}\log \pi_\theta(y_t \mid x, p, y_{<t})\right]. \quad (1)$$

For the first reference page, we use mixed supervision: body regions are supervised in LaTeX, while detected reference regions are supervised in BibTeX entries, encouraging a clean transition from body generation to bibliography generation near the end of a paper.

5.2 Stage II: RLVR with Unit-Test Reward

While SFT teaches the model to imitate page labels, it does not directly optimize for *functional* correctness (e.g., whether a formula renders correctly, whether a table preserves numeric content and structure, or whether cross-references resolve). Inspired by olmOCR2 (Poznanski et al., 2025b), which leverages unit-test pass rates as a verifiable training signal for document OCR, we therefore apply RLVR on single-page inputs. For each page x, we sample a group of K completions $\{y^{(k)}\}_{k=1}^{K} \sim \pi_\theta(\cdot \mid x, p)$ and score each completion using a suite of automatically constructed *binary unit tests*. The page-level reward is the fraction of unit tests that pass:

$$R(x, y) = \frac{1}{|\mathcal{T}(x)|}\sum_{\tau\in\mathcal{T}(x)} \mathbb{1}[\tau(y) = \texttt{pass}] \in [0, 1], \quad (2)$$

where $\mathcal{T}(x)$ denotes the test set instantiated for page x from its aligned ground truth and auxiliary layout signals. We optimize the policy with a group-based RL objective (e.g., GRPO-style updates) using within-group normalization for variance reduction, and a KL penalty to keep the learned policy close to the SFT reference policy π_{ref}:

$$\max_\theta \mathbb{E}_x\left[\frac{1}{K}\sum_{k=1}^{K}\left(\hat{A}^{(k)} \cdot \log \pi_\theta(y^{(k)} \mid x, p)\right)\right] - \beta\,\mathbb{E}_x[\mathrm{KL}(\pi_\theta(\cdot \mid x, p) \,\|\, \pi_{\text{ref}}(\cdot \mid x, p))]. \quad (3)$$

where $\hat{A}^{(k)}$ is a group-relative advantage derived from the rewards $\{R(x, y^{(k)})\}_{k=1}^{K}$.

Unit-Test Reward Design. We construct the page-level verifiable reward by adapting the full TEXOCR-Bench evaluation suite (see Section 3.1) to single-page scoring. For each page, we reformulate all **nine metrics** into deterministic binary (pass/fail) unit tests covering the three metric groups: (i) *transcription-fidelity tests* check faithful content recovery, including complex text preservation on extracted plain-text anchor sentences, table accuracy via numerical-entry overlap, and formula accuracy via token-level equivalence after normalization; (ii) *structural-faithfulness tests* probe document-level invariants, including section-hierarchy consistency, citation-key validity against the page's bibliography, and label–reference resolution for figures and tables; (iii) *end-to-end usability*

Method	Structural Faithfulness				End-to-End Usability				Transcription Fidelity				Overall
	SA	CC	RV	Avg	DS	Baseline	CSR	Avg	CTP	FA	TA	Avg	
GPT-5.3	**83.4**	**86.0**	65.2	78.2	**72.4**	**98.8**	**82.7**	**84.6**	69.2	57.9	**91.1**	72.7	78.5
TEXOCR (SFT + RLVR)	76.7	85.9	**86.8**	**83.1**	64.7	95.2	45.2	68.4	72.8	58.4	89.2	73.5	75.0
TEXOCR (SFT)	73.5	74.5	74.1	74.0	61.8	91.8	44.3	66.0	69.0	53.3	88.0	70.1	70.0
Qwen3-VL-32B	61.0	62.6	42.9	55.5	67.0	98.3	58.9	74.7	75.7	63.8	88.9	**76.1**	68.8
Qwen3-VL-8B	72.9	35.6	9.7	39.4	66.1	98.7	59.0	74.6	**77.2**	62.2	77.0	72.1	62.2
Mistral-Small-3.1-24B	68.9	43.2	44.6	52.2	60.4	98.4	39.1	66.0	62.1	47.4	79.3	62.9	60.4
Infinity-Parser-7B	77.5	3.7	17.6	32.9	68.9	96.5	44.5	70.0	76.8	**76.1**	80.8	74.6	59.1
Qwen2.5-VL-32B	68.0	15.9	8.6	30.8	62.1	96.2	55.8	71.4	67.6	62.7	81.9	70.7	57.6
Qwen3-VL-4B	64.2	17.3	8.5	30.0	63.1	98.2	58.4	73.2	69.1	58.7	52.6	60.1	54.5
Qwen3-VL-2B	49.6	21.8	1.5	24.3	50.3	97.9	57.4	68.5	67.3	55.2	68.8	63.8	52.2
Qwen2.5-VL-7B	53.9	4.8	9.2	22.6	52.3	94.6	58.2	68.4	56.8	53.2	51.1	53.7	48.2
olmOCR-2-7B	43.7	0.5	0.2	14.8	65.9	96.1	36.5	66.2	74.4	60.5	49.3	61.4	47.5
Pixtral-12B	52.1	4.8	2.1	19.7	50.8	92.7	64.5	69.3	47.9	42.7	62.8	51.1	46.7
InternVL3-38B	58.3	21.1	10.4	29.9	40.8	92.3	39.6	57.6	26.3	38.2	61.3	41.9	43.1
InternVL3-8B	51.9	17.0	3.6	24.2	49.1	93.1	35.1	59.1	46.6	37.6	50.4	44.9	42.7
InternVL2.5-38B	54.6	13.0	8.1	25.2	45.1	88.9	42.7	58.9	36.9	35.0	43.1	38.3	40.8
Qwen2.5-VL-3B	52.2	1.3	0.9	18.1	43.2	76.9	39.5	53.2	43.2	43.8	45.2	44.1	38.5
DeepSeek-OCR	4.1	0.3	0.0	1.5	33.8	94.7	50.1	59.5	44.3	48.3	2.0	31.5	30.8
Phi-4-multimodal	18.6	7.7	6.2	10.8	19.2	98.3	56.8	58.1	1.5	26.6	5.4	11.2	26.7
InternVL2.5-8B	35.0	5.8	2.0	14.3	26.2	73.3	36.8	45.4	7.4	28.1	20.8	18.8	26.2
LLaVA-OneVision	26.4	2.2	0.6	9.7	22.7	65.4	46.1	44.7	6.1	26.8	12.6	15.2	23.2
InternVL2-8B	36.9	5.1	1.6	14.5	24.4	61.5	24.4	36.8	6.0	28.0	20.8	18.3	23.2
Phi-3.5-vision	22.5	1.2	1.1	8.3	15.6	87.7	44.9	49.4	2.5	26.6	4.8	11.3	22.9

Table 2: Main results on TEXOCR-Bench, reporting scores across three metric groups along with the Overall score: **Structural Faithfulness** — Section Accuracy (SA), Citation Coverage (CC), and Reference Validity (RV); **End-to-End Usability** — Document-Level Similarity (DS), page-level baseline sanity check (Baseline), and Compilation Success Rate (CSR); and **Transcription Fidelity** — Complex Text Preservation (CTP), Formula Accuracy (FA), and Table Accuracy (TA). The best scores per column are in **bold** and second-best are underlined.

tests target executable quality through a page-level sanity check that rejects empty, ill-formed, or degenerately repetitive outputs, a document-similarity test based on normalized edit distance, and a compilation probe that wraps the generated snippet in a minimal preamble and invokes a standard LaTeX compiler. For continuous metrics, we binarize by thresholding so that each test produces a clean pass/fail outcome; the scalar reward is then the fraction of tests passed.

6 Experiments

We benchmark a broad set of frontier models on TEXOCR-Bench to assess the state of PDF-to-LaTeX reconstruction, analyze common failure modes, and validate the effectiveness of our training recipe through systematic ablations.

6.1 Experimental Setup

We evaluate a broad set of contemporary MLLMs on TEXOCR-Bench under a unified page-level inference protocol.

Models. We evaluate a diverse collection of frontier models that support both visual and textual inputs. On the open-source side, we benchmark **thirteen open-source model series**: InternVL-2/2.5/3 (Chen et al., 2024b,a), Qwen2.5-VL (Bai et al., 2025a) and Qwen3-VL (Team, 2025), DeepSeek-OCR (Wei et al., 2025), olmOCR-2-7B (Poznanski et al., 2025b), Infinity-Parser-7B (Wang et al., 2025), Pixtral (Agrawal et al., 2024), Mistral-Small-3.1 (Mistral AI, 2025), LLaVA-OneVision (Li et al., 2024), Phi-3.5-Vision, and Phi-4-Multimodal (Microsoft, 2024; Abouelenin et al., 2025). On the proprietary side, we include **OpenAI GPT-5.3** as the representative closed-source baseline. For open-source models, we run inference with vLLM (Kwon et al., 2023) to ensure efficient, reproducible decoding; the proprietary model is accessed via the official API using recommended default settings. The model prompt is provided in Appendix C. Detailed model configurations are provided in Appendix D.

TEXOCR Training. We fine-tune a **Qwen3-VL-2B** base model on our dataset using SFT. The model is trained with full-parameter updates for 3 epochs at a learning rate of $1e-5$. Each training sample pairs a rendered PDF page image with its corresponding LaTeX source.

Error Type	Description
Paragraph Truncation and Missing Text	Partial paragraphs are often omitted at page beginnings or endings, especially near page boundaries or section transitions, leading to systematic loss of textual fragments.
Mathematical Formula Mismatches	Formulas frequently contain incorrect symbols, missing or mismatched delimiters, or improper inline/display classification, resulting in invalid or non-compilable expressions.
Table Structure Corruption	Tables suffer from structural errors such as missing column separators, misaligned rows, or incorrect multi-line cell handling, which easily break table environments.
Citation and Reference Errors	Citations are sometimes omitted or generated with incorrect formats, causing unresolved `\cite` or `\ref` commands and broken cross-referencing.
Compilation Failures	Malformed formulas, corrupted tables, and reference errors often lead to LaTeX compilation failures, significantly limiting practical usability.

Table 3: Common error types observed in PDF-to-LaTeX generation and their typical manifestations.

Inference Protocol. We adopt a unified page-level inference protocol: each PDF page is rendered as a single image and processed independently, and the resulting LaTeX outputs are concatenated in document order for evaluation. This page-level setting is a necessary compromise—multi-image and merged multi-page alternatives we investigated suffer from cross-page interference and resolution loss, respectively, as detailed in Section 6.4.

Evaluation Metrics. We report the full nine-metric suite introduced in Section 3.1, organized along the three dimensions defined there. Unless otherwise specified, page-level outputs are first merged to form document-level predictions, and metrics are then computed at the document level and averaged across all documents.

6.2 Main Findings

Table 2 reports results across all models, yielding three observations. First, proprietary MLLMs are the most consistent: they preserve complex body text and remain strong on formulas and tables, while many open-source models handle plain text well but degrade on syntax-sensitive tasks with malformed equations, broken table structures, and inconsistent reference formatting. Second, our two-stage training recipe is effective: SFT alone already delivers large gains over the Qwen3-VL-2B base across all three dimensions, and adding RLVR further improves aspects poorly captured by token-level likelihood—particularly structural faithfulness and end-to-end usability—while preserving SFT's transcription quality. Third, compilation success is a stricter usability indicator than recognition metrics alone: many generated projects still fail to compile without manual correction, typically due to unclosed math environments, malformed tables, and inconsistent `\ref{...}` labels, revealing that current models lack sufficient syntactic robustness and project-level consistency for reliable downstream use.

Model	Single-Image	Multi-Image	Merged
Qwen3VL-2B	52.2	39.1	36.9
GPT-5.3	78.5	56.9	42.6

Table 4: Overall score on TEXOCR-Bench under single-image, multi-image, and merged inference strategies.

6.3 Error Analysis and Case Study

To understand failure modes beyond aggregate metrics, we conduct a targeted error analysis on our evaluation set. Our analysis reveals several recurring error patterns that directly affect the completeness, correctness, and compilability of the generated LaTeX. We summarize these errors in Table 3 and illustrate each category with representative case studies in Appendix E. A particularly extreme failure mode is exemplified by DeepSeek-OCR: regardless of how we modify the prompt, the model consistently produces Markdown-style linearized output rather than LaTeX, leaving virtually no valid section, citation, reference, or `tabular` commands for the evaluator to score. This accounts for its near-zero scores on structural faithfulness and table accuracy in Table 2, despite competitive performance on coarser transcription checks.

6.4 Effect of Inference Granularity

We compare single-image inference with multi-image and merged multi-page inference strategies for PDF-to-LaTeX conversion. As shown in Table 4, single-image inference consistently achieves better performance. Multi-image inference introduces cross-page interference that degrades vi-

Method	Structural Faithfulness				End-to-End Usability				Transcription Fidelity				Overall
	SA	CC	RV	Avg	DS	Baseline	CSR	Avg	CTP	FA	TA	Avg	
SFT+RLVR	76.7	85.9	86.8	83.1	64.7	95.2	45.2	68.4	72.8	58.4	89.2	73.5	75.0
w.o. Transcription Fidelity	75.0	76.5	80.6	77.4	60.3	93.3	48.9	67.5	65.6	53.4	87.7	68.9	71.3
w.o. Structural Faithfulness	74.8	69.4	75.1	73.1	61.6	93.5	46.4	67.2	68.6	54.6	88.5	70.6	70.3
w.o. End-to-End Usability	75.2	72.5	77.6	75.1	60.2	93.0	46.3	66.5	67.7	54.2	88.3	70.1	70.5

Table 5: Ablation of verifiable unit tests in RLVR, where removing a specific test leads to a clear degradation in corresponding evaluation metric, highlighting the targeted supervision provided by each unit-test reward.

sual–text alignment, while merging multiple pages into a single image leads to resolution loss and visual clutter, both of which harm structural accuracy. In contrast, single-image inference preserves page-level locality and visual clarity, resulting in more accurate and stable LaTeX generation.

6.5 Ablation Study

We conduct a set of controlled ablation studies to isolate the contributions of different training components and design choices in our framework.

Ablation of Verifiable Unit Tests. Next, we ablate individual categories of verifiable unit tests in RLVR to assess their impact. As shown in Table 5, removing a given test consistently degrades its corresponding evaluation metric (e.g., lower formula accuracy without formula tests), with little compensation from other metrics. This highlights the targeted role of unit-test rewards: each test explicitly supervises a specific functional capability, and its absence directly weakens that behavior. Overall, these results show that gains in structural faithfulness and end-to-end usability require explicit, verifiable objectives rather than emerging implicitly.

Effect of Group Size K. Finally, we analyze the effect of group size K in RLVR for $K \in \{4, 8, 12, 16, 20, 24\}$. Larger K consistently yields more stable performance across metrics, while small K leads to higher variance and less reliable gains. Increasing K reduces optimization noise and produces smoother improvements, indicating more accurate relative advantage estimates and stronger variance reduction. Overall, sufficiently large group sizes are crucial for stable training and for fully realizing the benefits of RLVR.

7 Conclusion

This work targets a practical gap in PDF-to-LaTeX reconstruction: generating output that not only looks correct, but also compiles and preserves document structure reliably. We introduce TEXOCR-Bench to evaluate executable, structure-faithful reconstruction at scale, and TEXOCR-Train to enable training with page-aligned supervision. Using unit-test style, verifiable rewards, we show that optimizing directly for functional correctness reduces common failure modes such as broken environments, malformed tables, and invalid references. These resources and results establish a stronger baseline for developing OCR models that can recover scientific documents into usable LaTeX, and provide actionable insights for future advancements.

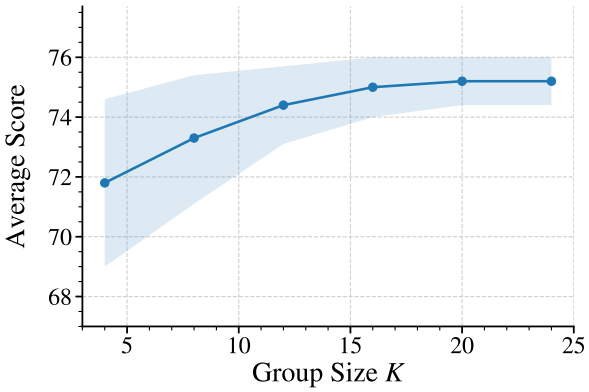

Figure 2: Effect of group size K in RLVR. We report the average benchmark score as a function of K, with shaded regions indicating the standard deviation across evaluation samples.

Limitation

This work provides a first step toward reliable page-level reconstruction of scientific documents into compilable LaTeX. While our results demonstrate strong performance under a page-wise inference setting, the overall task is inherently document-centric. An important direction for future work is to move beyond isolated page-level reconstruction and develop more effective document-level approaches that better capture cross-page structure, global consistency, and long-range dependencies.

References

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, and 1 others. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*.

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khandelwal, Timothée Lacroix, Guillaume Lample, Diego Las Casas, Thibaut Lavril, and 23 others. 2024. Pixtral 12b. *Preprint*, arXiv:2410.07073.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025a. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Weikang Bai, Yongkun Du, Yuchen Su, Yazhen Xie, and Zhineng Chen. 2025b. Complex mathematical expression recognition: Benchmark, large-scale dataset and strong baseline. *Preprint*, arXiv:2512.13731.

Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. 2023. Nougat: Neural optical understanding for academic documents. *Preprint*, arXiv:2308.13418.

Qian Chen, Xianyin Zhang, Lifan Guo, Feng Chen, and Chi Zhang. 2025a. Dianjin-ocr-r1: Enhancing ocr capabilities via a reasoning-and-tool interleaved vision-language model. *Preprint*, arXiv:2508.13238.

Song Chen, Xinyu Guo, Yadong Li, Tao Zhang, Mingan Lin, Dongdong Kuang, Youwei Zhang, Lingfeng Ming, Fengyu Zhang, Yuran Wang, Jianhua Xu, Zenan Zhou, and Weipeng Chen. 2025b. Ocean-ocr: Towards general ocr application via a vision-language model. *Preprint*, arXiv:2501.15558.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024a. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, and 1 others. 2024b. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*.

Christopher Clark and Santosh Divvala. 2016. Pdffigures 2.0: Mining figures from research papers. In *2016 IEEE/ACM Joint Conference on Digital Libraries (JCDL)*, pages 143–152.

Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, Yue Zhang, Yubo Zhang, Handong Zheng, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. 2025. Paddleocr-vl: Boosting multilingual document parsing via a 0.9b ultra-compact vision-language model. *Preprint*, arXiv:2510.14528.

Daxiang Dong, Mingming Zheng, Dong Xu, Chunhuan Luo, Bairong Zhuang, Yuxuan Li, Ruoyun He, Haoran Wang, Wenyu Zhang, Wenbo Wang, Yicheng Wang, Xuehan Xiong, Ayong Zheng, Xiao gang Zuo, Zi-Wei Ou, Jing Gu, Quan gui Guo, Jianmin Wu, Dawei Yin, and Dou Shen. 2026. Qianfan-ocr: A unified end-to-end model for document intelligence.

Shuaiqi Duan, Ya-Qi Xue, Weihan Wang, Zhèngyuān Sū, Huan Liu, Shenghe Yang, Guobing Gan, Guo Wang, Zihan Wang, Sheng Yan, Dexin Jin, Yuxuan Zhang, Guohong Wen, Yanfeng Wang, Yutao Zhang, Xiaohan Zhang, Wenyi Hong, Yukuo Cen, Da Yin, and 4 others. 2026. Glm-ocr technical report.

Gavin Greif, Niclas Griesshaber, and Robin Greif. 2025. Multimodal llms for ocr, ocr post-correction, and named entity recognition in historical documents. *Preprint*, arXiv:2504.00414.

Ahmed Heakl, Muhammad Abdullah Sohail, Mukul Ranjan, Rania Elbadry, Ghazi Shazan Ahmad, Mohamed El-Geish, Omar Maher, Zhiqiang Shen, Fahad Shahbaz Khan, and Salman Khan. 2025. KITAB-bench: A comprehensive multi-domain benchmark for Arabic OCR and document understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22006–22024, Vienna, Austria. Association for Computational Linguistics.

Kotaro Inoue. 2025. Context-independent ocr with multimodal llms: Effects of image resolution and visual complexity. *Preprint*, arXiv:2503.23667.

Nan Jiang, Shanchao Liang, Chengxiao Wang, Jiannan Wang, and Lin Tan. 2025. Latte: Improving latex recognition for tables and formulae with iterative refinement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4030–4038.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024. Llava-onevision: Easy visual task transfer. *Preprint*, arXiv:2408.03326.

Yumeng Li, Guang Yang, Hao Liu, Bowen Wang, and Colin Zhang. 2025a. dots.ocr: Multilingual document layout parsing in a single vision-language model. *Preprint*, arXiv:2512.02498.

Zichao Li, Aizier Abulaiti, Yaojie Lu, Xuanang Chen, Jia Zheng, Hongyu Lin, Xianpei Han, Shanshan Jiang, Bin Dong, and Le Sun. 2025b. READoc: A unified benchmark for realistic document structured extraction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21889–21905, Vienna, Austria. Association for Computational Linguistics.

Jun Ling, Yao Qi, Tao Huang, Shibo Zhou, Yanqin Huang, Jiang Yang, Ziqi Song, Ying Zhou, Yang Yang, Heng Tao Shen, and Peng Wang. 2025. Table2latex-rl: High-fidelity latex code generation from table images via reinforced multimodal language models. *Preprint*, arXiv:2509.17589.

Yuan Liu, Zhongyin Zhao, Le Tian, Haicheng Wang, Xubing Ye, Yangxiu You, Zilin Yu, Chuhan Wu, Zhou Xiao, Yang Yu, and Jie Zhou. 2025. POINTS-reader: Distillation-free adaptation of vision-language models for document conversion. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1576–1601, Suzhou, China. Association for Computational Linguistics.

Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel Weld. 2020. S2ORC: The semantic scholar open research corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4969–4983, Online. Association for Computational Linguistics.

Souvik Mandal, Ashish Talewar, Siddhant Thakuria, Paras Ahuja, and Prathamesh Juvatkar. 2025. Nanonets-ocr2: A model for transforming documents into structured markdown with intelligent content recognition and semantic tagging.

Microsoft. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Mistral AI. 2025. Mistral-small-3.1-24b-instruct-2503. `https://huggingface.co/mistralai/Mistral-Small-3.1-24B-Instruct-2503`. Apache 2.0 License.

Ahmed Nassar, Andres Marafioti, Matteo Omenetti, Maksym Lysak, Nikolaos Livathinos, Christoph Auer, Lucas Morin, Rafael Teixeira de Lima, Yusik Kim, A. Said Gurbuz, Michele Dolfi, Miquel Farré, and Peter W. J. Staar. 2025. Smoldocling: An ultra-compact vision-language model for end-to-end multi-modal document conversion. *Preprint*, arXiv:2503.11576.

OpenAI. 2025. Gpt-5 is here.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. 2025. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. *Preprint*, arXiv:2412.07626.

PDF Association staff. 2015. Pdf in 2016: Broader, deeper, richer. *PDF Association*.

Jake Poznanski, Aman Rangapur, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Aman Rangapur, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. 2025a. olmocr: Unlocking trillions of tokens in pdfs with vision language models. *Preprint*, arXiv:2502.18443.

Jake Poznanski, Luca Soldaini, and Kyle Lo. 2025b. olmocr 2: Unit test rewards for document ocr. *Preprint*, arXiv:2510.19817.

Satadal Saha, Subhadip Basu, Mita Nasipuri, and Dipak Kr. Basu. 2010. A hough transform based technique for text segmentation. *Preprint*, arXiv:1002.4048.

Said Taghadouini, A. Cavaillès, and Baptiste Aubertin. 2026. Lightonocr: A 1b end-to-end multilingual vision-language model for state-of-the-art ocr. *ArXiv*, abs/2601.14251.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Baode Wang, Biao Wu, Weizhen Li, Meng Fang, Zuming Huang, Jun Huang, Haozhe Wang, Yanjie Liang, Ling Chen, Wei Chu, and Yuan Qi. 2025. Infinity parser: Layout aware reinforcement learning for scanned document parsing. *Preprint*, arXiv:2506.03197.

Bin Wang, Tianyao He, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Tao Chu, Yuan Qu, Zhenjiang Jin, Weiju Zeng, Ziyang Miao, Bangrui Xu, Junbo Niu, Mengzhang Cai, Jiantao Qiu, Qintong Zhang, Dongsheng Ma, Yuefeng Sun, Hejun Dong, Wenzheng Zhang, and 24 others. 2026. Mineru2.5-pro: Pushing the limits of data-centric document parsing at scale.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han, and Xiangyu Zhang. 2024. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. *Preprint*, arXiv:2409.01704.

Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.

Haoran Wei, Yaofeng Sun, and Yukun Li. 2026. Deepseek-ocr 2: Visual causal flow. *ArXiv*, abs/2601.20552.

Christoph Wick, Christian Reul, and Frank Puppe. 2018. Calamari - a high-performance tensorflow-based deep learning package for optical character recognition. *Preprint*, arXiv:1807.02004.

Hao Wu, Haoran Lou, Xinyue Li, Zuodong Zhong, Zhaojun Sun, Phellon Chen, Xuanhe Zhou, Kai Zuo, Yibo Chen, Xu Tang, Yao Hu, Boxiang Zhou, Jian Wu, Yongji Wu, Wenxin Yu, Yingmiao Liu, Yuhao Huang, Manjie Xu, Gang Liu, and 3 others. 2026. Firered-ocr technical report.

Handong Zheng, Yumeng Li, Kaile Zhang, Liang Xin, Guan yu Zhao, Hao Liu, Jiayu Chen, Jie Lou, Jiyu Qiu, Qingshun Fu, Rui Yang, Shuo Jiang, W. Luo, Weijie Su, Weijun Zhang, Xingyu Zhu, Yabin Li, Yiwei Ma, Yu Chen, and 6 others. 2026. Multimodal ocr: Parse anything from documents.

A Implementation Details of Evaluation Metrics

This appendix provides a detailed description of the implementation of each evaluation metric used in our benchmark. All metrics are computed deterministically based on the generated LaTeX output and the corresponding ground-truth LaTeX source. Unless otherwise stated, evaluation is performed at the document level by aggregating page-level outputs in their original order.

A.1 Complex Text Preservation

To assess whether complex body text is preserved, we extract one representative plain-text sentence from each section of the ground-truth LaTeX source. Sentences containing LaTeX commands or environments are excluded to avoid trivial mismatches. For each extracted sentence, we perform an exact substring match against the concatenated generated LaTeX output. The context existence score is defined as the ratio of ground-truth sentences that are successfully recovered in the generated text.

A.2 Document-Level Similarity

We measure overall textual similarity between the generated LaTeX and the ground-truth source using normalized Levenshtein distance. Before comparison, BibTeX entries are removed from the generated output to avoid penalizing formatting differences in references. The similarity score is computed as

$$\text{Similarity} = 1 - \frac{\text{Levenshtein}(A, B)}{\max(|A|, |B|)},$$

where A and B denote the generated and reference LaTeX strings, respectively.

A.3 Table Accuracy

Table accuracy evaluates whether tabular content is correctly reconstructed. For each reference table, we identify the corresponding generated table on the same page. We extract numerical entries (including percentages) from the longest `tabular` environment and normalize them into numeric tokens. Two tables are considered a match if they satisfy either: (i) a high overlap ratio between numerical entries; or (ii) a moderate overlap ratio combined with a high hit rate on unique numerical anchors (numbers that appear exactly once in the reference table). This metric is binary per table. The final table accuracy is computed as the proportion of reference tables that are successfully matched.

A.4 Formula Accuracy

Formula accuracy measures whether mathematical expressions are faithfully transcribed. We extract display math environments (e.g., `equation` and `eqnarray`) from both generated and reference LaTeX. Each generated formula is aligned to at most one reference formula using sequence similarity after normalization, which removes comments, layout commands (e.g., `\left`, `\right`), and whitespace. Alignment is accepted if the similarity exceeds a fixed threshold. For matched formula pairs, we further verify equivalence using token-level comparison: formulas are tokenized into LaTeX commands, variables, numbers, and operators, and are considered correct if the token sequences are identical or one is an ordered subsequence of the other. Formula accuracy is reported as the fraction of reference formulas that are correctly matched.

A.5 Baseline Validity Check

The baseline metric serves as a coarse-grained sanity check to detect severe page-level generation failures. For each generated page, we verify that the output satisfies a set of minimal usability constraints: (i) the output is a non-empty string; (ii) it contains at least one alphanumeric character; (iii) it does not contain non-target language characters (e.g., CJK characters); (iv) it does not contain emoji symbols; and (v) it does not exhibit degenerate repetition patterns, defined as repeated trailing n-grams for any n up to a fixed threshold. A page is counted as valid if it passes all checks. The baseline score is computed as the fraction of valid pages among all pages in the document.

A.6 Section Accuracy

Section accuracy evaluates whether structural section headers are correctly recovered. We extract all occurrences of `\section`, `\subsection`, and `\subsubsection` commands from both the generated and reference LaTeX. Section titles are normalized by removing numeric prefixes (e.g., "3.2"). A predicted section is considered correct if its title matches a reference title under bidirectional substring inclusion. Each predicted section can be matched to at most one ground-truth section. We

report precision as the fraction of predicted sections that are successfully matched.

A.7 Citation Coverage

Citation coverage evaluates whether in-text citation commands are correctly generated. We extract all citation commands (e.g., `\cite`, `\citep`, `\citet`) and their keys from both the generated output and the reference source. A generated citation key is considered valid if it either refers to a numeric citation index within the number of BibTeX entries or appears in the concatenated BibTeX content of the generated document. Citation precision is computed as the ratio of valid generated citations to the total number of ground-truth citations.

A.8 Figure and Table Reference Validity

To evaluate cross-reference correctness, we extract all `\ref{...}` commands from both the generated and reference LaTeX. For each figure and table label defined in the reference, we count the number of times it is referenced. A label is considered correct if the generated output references it the same number of times as in the ground truth. Label precision is computed as the fraction of labels that satisfy this condition.

B OCR Benchmark Construction

B.1 PDF2Latex Instruction

PDF-to-LaTeX Annotation Instructions (Data with LaTeX Sources)

You are given the original LaTeX source. Treat it as the source of truth. You are not rewriting content. Your task is to make it compile cleanly in our template while keeping the meaning unchanged.
Do not drop any visible text, equations, tables, captions, or partial sentences at page boundaries. Do not invent missing content. Keep math symbols, subscripts, superscripts, and indices exactly as they appear.
You may do light cleanup that does not change meaning, such as removing LaTeX comments, fixing whitespace, escaping special characters, and ensuring that braces and environments are balanced. Use standard environments such as `\section`, `\equation` or `align`, `table` and `tabular`, and `figure`. For figures, use an `\includegraphics` placeholder and always include a caption and a label. For tables, keep the same column count and row breaks, and copy all numbers exactly.
Convert citations into `\cite{}` and do not invent citation keys. Follow the dataset citation schema. If the source uses numeric citations, represent them as `\cite{1,2}`. If the source uses author-year citations, map them to keys that match the provided BibTeX entries. Do not change which sentences the citations attach to.
Labels and references must be standardized and consistent. Use `fig:figure_1`, `fig:figure_2`, and so on for figures, and `tab:table_1`, `tab:table_2`, and so on for tables. Use `eq:equation_1`, `eq:equation_2`, and so on for displayed equations when an equation label is needed. Always reference them with `\ref{}` or `\eqref{}` using the same label you defined. If the text says "Figure 3" or "Table 2", make sure the label you assign matches that numbering, and update all references to match.

PDF-to-LaTeX Annotation Instructions (Data without LaTeX Sources)

You are given a PDF but not the original LaTeX source. Reconstruct the document as faithful and compilable LaTeX in our template. Within that constraint, match the original wording, math, and structure as closely as possible.
Avoid paraphrasing unless it is necessary to make the output compile. Use the same conventions as above for environments, figures, and tables. Figures should be represented with an `\includegraphics` placeholder and must include a caption and a label. Tables should be rebuilt with the correct grid, separators, and row breaks, and you must copy all numeric entries exactly. If a table is extremely complex, prioritize compilability while preserving all numeric entries and any structure you can reliably recover.
For citations, keep them where they appear in the PDF. Convert them into `\cite{}` without inventing keys and follow the dataset citation schema. Numeric citations should be written as `\cite{1,2}`. Author-year citations should be mapped to keys that match the provided BibTeX entries. Do not move citations to different sentences.
For labels and cross-references, use a standardized scheme with explicit prefixes. Use `fig:figure_1`, `tab:table_1`, and `eq:equation_1` style labels, and use `\ref{}` or `\eqref{}` consistently so cross-references resolve. Make sure the numbering is aligned with the PDF. If the PDF references "Fig. 4", you should label that figure as `fig:figure_4` and ensure every mention points to `\ref{fig:figure_4}`. If something in the PDF is unclear, do not guess. Keep it conservative, prioritize compilability, and preserve what is unambiguous.

B.2 Annotator Information

ID	Position	Disciplinary
1	Master Student	Computer Science
2	Master Student	Engineering
3	Master Student	Healthcare
4	Postdoc	Nature Science
5	Master Student	Engineering
6	Master Student	Computer Science
7	Master Student	Healthcare
8	PhD Student	Nature Science
9	Postdoc	Engineering
10	Master Student	Nature Science
11	PhD Student	Healthcare
12	Research Scientist	Computer Science
13	Master Student	Healthcare
14	Master Student	Engineering
15	PhD Student	Computer Science

Table 6: Information of annotators engaged in dataset construction. The average payment rate is 13 US dollars.

C Model Inference Prompt

Model Inference Prompt

Convert the content of a specific page from the provided PDF into LaTeX format, following these requirements:

1. **Convert only the main body text**: Do not include headers, footers, page numbers, or decorative page elements.
2. If the page is the **first page of the paper**, use `\title{}` for the title, `\author{}` for the author information (if any), and

   ```
   \begin{abstract}
   xxx
   \end{abstract}
   ```

   for the abstract, then begin the main text using `\section`.
3. **Do not use any packages or preamble**: Output only the LaTeX code corresponding to the body content. If the beginning of the page is the continuation of a previous section, convert that text as well without omission.
4. **Handling figures and tables**:
   - Name all figure files as `figure_x.pdf`, where `x` is the figure's ID from the original document.
   - If a figure contains multiple subfigures, convert it using subfigure format and name the files `figure_x_1.pdf`, `figure_x_2.pdf`, etc.
   - Assign figure labels using `\label{figure_x}`.
   - Identify table IDs similarly, and use `\label{table_x}` for tables.
   - If the page contains references to figures or tables, convert them according to these rules, for example `\ref{fig:figure_x}`, `\ref{tab:table_x}`.
   - You should use `\begin{figure}` to describe the figure.
5. **Correctly convert the section structure**:
   - Use `\section`, `\subsection`, etc., following the exact hierarchical structure of the original text.
6. **Handling citations**:
   - Convert numeric citations like `[1,2]` to `\cite{1,2}`.
   - Convert author-year citations like `(Li et al., 2023; Burns et al., 2022)` into `\cite{Li_2023, Burns_2022}`.
   - If there are two authors, such as "Pezeshkpour & Hruschka, 2023", convert it to `\cite{Pezeshkpour_Hruschka_2023}`, ensuring the citation key reflects the authors and year.
7. **Incomplete content at the top or bottom of the page**: Do not modify or complete it. Keep the original meaning and convert it as is.
8. **If the page contains part of the References section**:
   - Convert all reference entries on the page into BibTeX format only (e.g., `@article{Smith_2020, ...}`).
   - Do not use `\bibitem`.
   - The BibTeX entry keys must exactly match the citation labels generated in Rule 6.
   - If this page begins the References section, start with `\section{References}`.

Directly output the LaTeX code, without any extra output.

D Model Configuration

Organization	Model	Release	Version	# Inference Pipeline
		Proprietary Models		
OpenAI	GPT-5.3	2026-03	`gpt-5.3-chat-latest`	API
		Open-source Multimodal Foundation Models		
Alibaba	Qwen2.5-VL-32B	2025-01	`Qwen2.5-VL-32B-Instruct`	vLLM
	Qwen2.5-VL-7B	2025-01	`Qwen2.5-VL-7B-Instruct`	
	Qwen2.5-VL-3B	2025-01	`Qwen2.5-VL-3B-Instruct`	
	Qwen3-VL-2B	2025-10	`Qwen3-VL-2B-Instruct`	
	Qwen3-VL-4B	2025-10	`Qwen3-VL-4B-Instruct`	
	Qwen3-VL-8B	2025-10	`Qwen3-VL-8B-Instruct`	
	Qwen3-VL-32B	2025-10	`Qwen3-VL-32B-Instruct`	
Mistral AI	Mistral-Small-3.1	2025-03	`Mistral-Small-3.1-24B`	vLLM
	Pixtral-12B	2024-09	`Pixtral-12B-2409`	
Shanghai AI Lab	InternVL3-38B	2025-04	`InternVL3-38B`	vLLM
	InternVL3-8B	2025-04	`InternVL3-8B`	
	InternVL2.5-38B	2024-11	`InternVL2.5-38B`	
	InternVL2.5-8B	2024-11	`InternVL2.5-8B`	
	InternVL2-8B	2024-07	`InternVL2-8B`	
Microsoft	Phi-3.5-Vision	2024-07	`Phi-3.5-Vision-Instruct`	vLLM
	Phi-4-Multimodal	2025-03	`Phi-4-Multimodal`	
Llava Hugging Face	LLaVA-OneVision-7B	2024-09	`llava-onevision-qwen2-7b-ov-chat-hf`	vLLM
DeepSeek	DeepSeek-OCR	2025-10	`DeepSeek-OCR`	vLLM
Infly	Infinity-Parser-7B	2025-06	`Infinity-Parser-7B`	vLLM
Allenai	olmOCR-2-7B	2025-10	`olmOCR-2-7B-1025`	olmOCR toolkit

Table 7: Details of the multimodal foundation models evaluated in our study. Models are organized by organization and aligned with performance data from the main text.

E Error Analysis

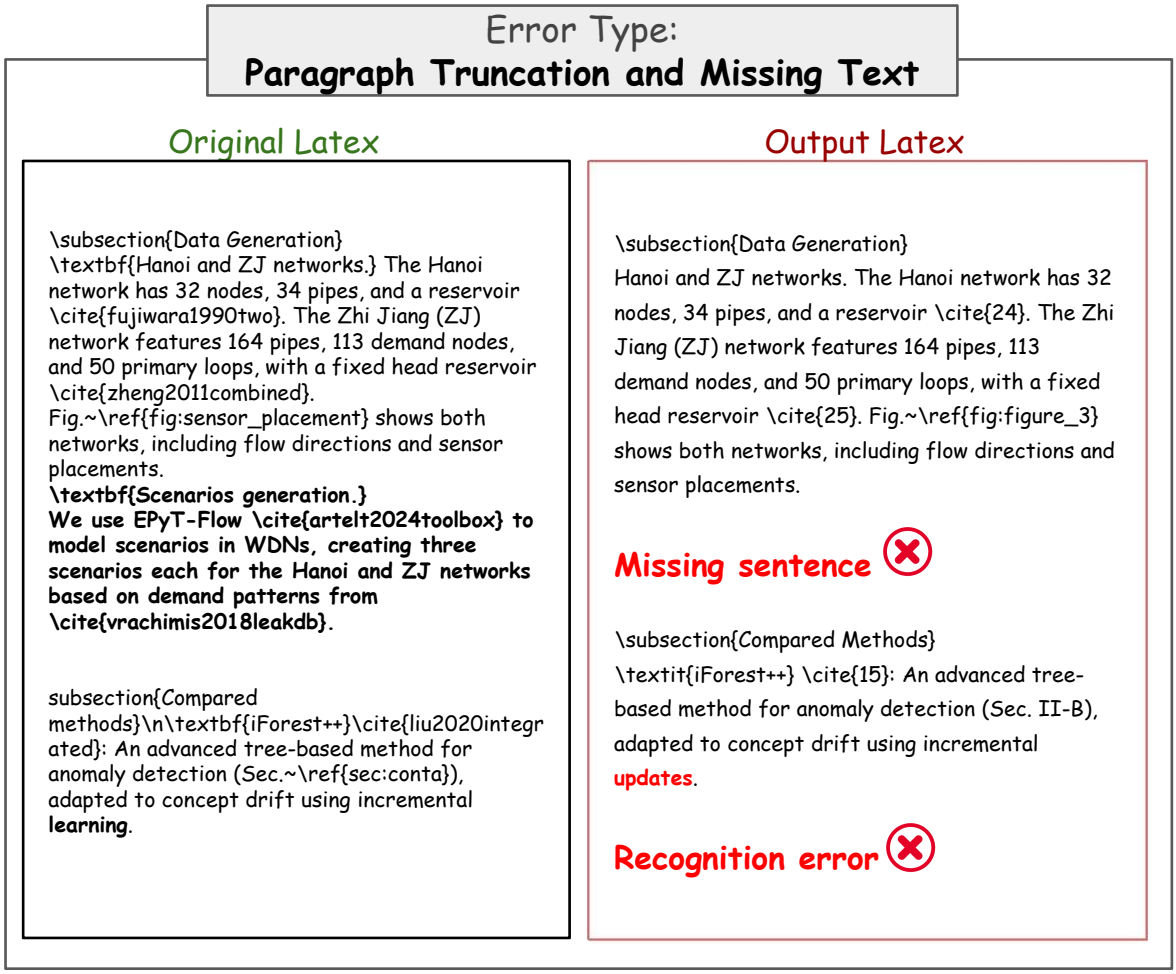

Figure 3: Example of paragraph truncation and missing text.

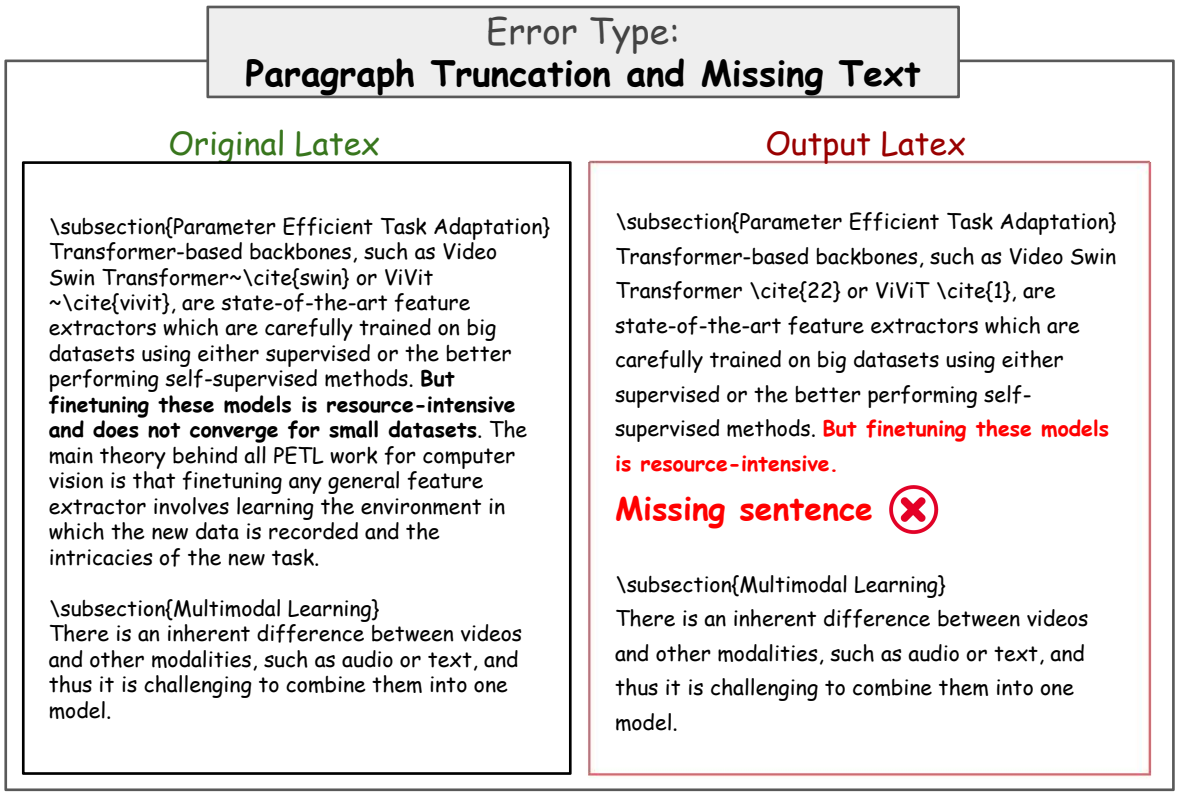

Figure 4: Example of paragraph truncation and missing text.

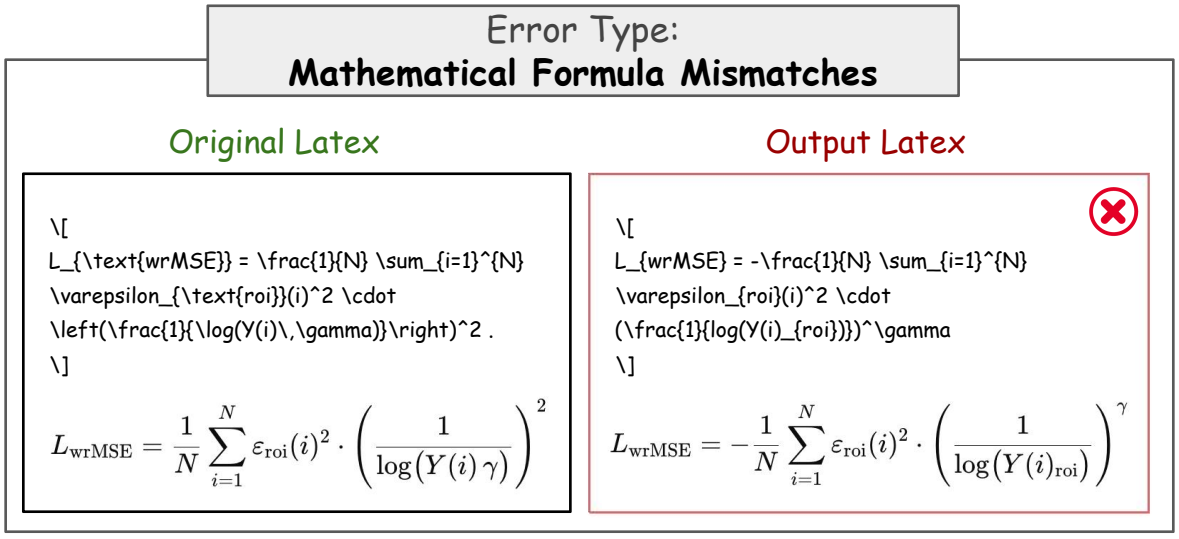

Figure 5: Example of mathematical formula mismatches.

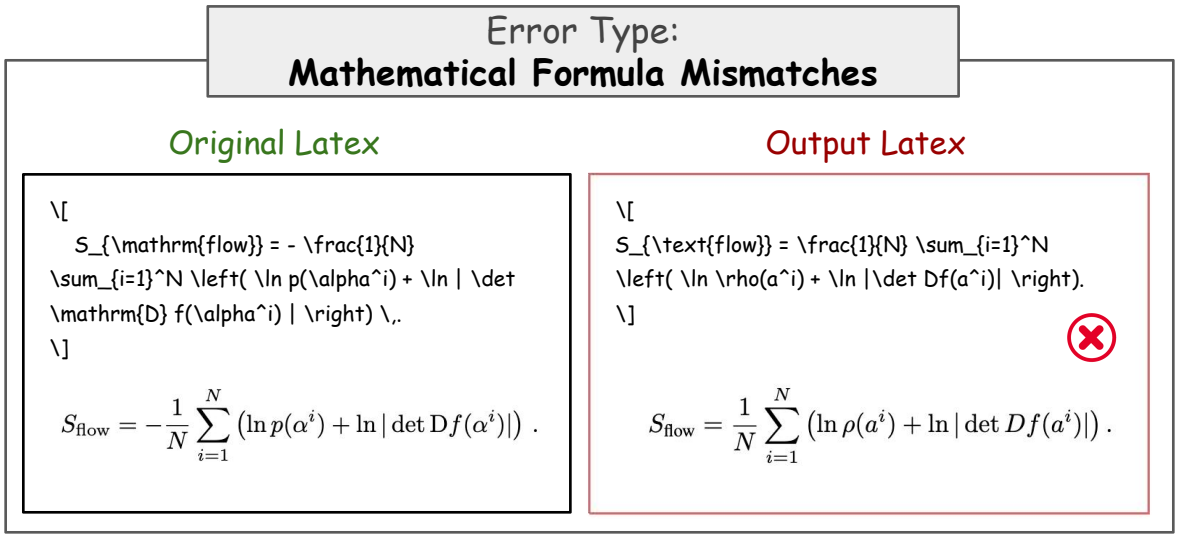

Figure 6: Example of mathematical formula mismatches.

Error Type:
Table Structure Corruption

Original Latex

Methods	Accuracy in each session (%)									Average
	0	1	2	3	4	5	6	7	8	Acc.
Topic	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62
CEC	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53
FACT	74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10	62.24
C-FSCIL	77.47	72.40	67.47	63.25	59.84	56.95	54.42	52.47	50.47	61.64
CLOM†	74.20	69.83	66.17	62.39	59.26	56.48	54.36	52.16	50.25	60.57
DBONet†	77.81	73.62	71.04	66.29	63.52	61.01	58.37	56.89	55.78	64.93
MCNet	73.30	69.34	65.72	61.70	58.75	56.44	54.59	53.01	50.72	60.40
SoftNet	72.62	67.31	63.05	59.39	56.00	53.23	51.06	48.83	46.63	57.57
WaRP	80.31	75.86	71.87	67.58	64.39	61.34	59.15	57.10	54.74	65.82
TEEN	74.92	72.65	68.74	65.01	62.01	59.29	57.90	54.76	52.64	63.10
NC-FSCIL	82.52	76.82	73.34	69.68	66.19	62.85	60.96	59.02	56.11	67.50
ALFSCIL	80.75	77.88	72.94	68.79	65.33	62.15	60.02	57.68	55.17	66.75
DyCR	75.73	73.29	68.71	64.80	62.11	59.25	56.70	54.56	52.24	63.04
baseline	73.92	67.14	63.71	60.07	57.10	54.85	52.52	50.49	48.60	58.71
baseline+ADBS	**79.93**	**75.22**	**71.11**	**65.99**	**62.46**	**58.38**	**55.96**	**53.72**	**51.15**	**63.77**
	(+6.02)	**(+8.08)**	**(+7.40)**	**(+5.92)**	**(+5.36)**	**(+3.53)**	**(+3.43)**	**(+3.22)**	**(+2.55)**	**(+5.06)**
OrCo*	79.77	63.29	62.39	60.13	58.76	56.56	55.49	54.19	51.12	60.19
OrCo+ADBS	79.77	**63.46**	61.89	**60.43**	**59.23**	56.32	**55.76**	**54.48**	**51.54**	**60.32**
	(+0.00)	**(+0.17)**	(-0.50)	**(+0.29)**	**(+0.46)**	(-0.25)	**(+0.27)**	**(+0.30)**	**(+0.42)**	**(+0.13)**
ALICE†*	80.37	72.34	67.67	63.61	61.11	58.53	57.40	55.43	53.46	63.32
ALICE+ADBS	80.12	**74.11**	**70.51**	**66.72**	**63.90**	**61.25**	**60.00**	**58.07**	**56.00**	**65.63**
	(-0.25)	**(+1.77)**	**(+2.84)**	**(+3.11)**	**(+2.79)**	**(+2.72)**	**(+2.60)**	**(+2.64)**	**(+2.54)**	**(+2.31)**
SAVC*	78.60	72.95	68.73	64.59	61.41	58.46	56.29	54.40	52.19	63.07
SAVC+ADBS	**85.13**	**80.39**	**77.07**	**72.61**	**69.54**	**66.54**	**64.70**	**62.72**	**60.60**	**71.03**
	(+6.53)	**(+7.43)**	**(+8.34)**	**(+8.03)**	**(+8.13)**	**(+8.08)**	**(+8.41)**	**(+8.32)**	**(+8.41)**	**(+7.96)**

Table 1. Comparison with SOTA methods on CIFAR100 for FSCIL. †: Boundary-based method. *: Reproduced results.

Output Latex

Table 1. Comparison with SOTA methods on CUB200 dataset for FSCIL. †: Boundary-based method. *: Reproduced results.

Methods	0	1	2	3	4	5	6	7	8	9	10	Average Acc.
Topic	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28	43.92
CEC	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33
FACT	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	64.42
CLOM†	79.57	76.07	72.94	69.82	67.80	65.56	63.94	62.59	60.62	60.34	59.58	67.17
DBONet†	78.66	75.53	72.72	69.45	67.21	65.15	63.03	61.77	59.77	59.01	57.42	66.34
MCNet	77.57	73.96	70.47	65.81	66.16	63.81	62.09	61.82	60.41	60.09	59.08	65.57
SoftNet	78.11	74.51	71.14	62.27	65.14	62.27	60.77	59.03	57.13	56.77	56.28	63.95
WaRP	77.74	74.15	70.82	66.90	65.01	62.64	61.40	59.86	57.95	57.77	57.01	64.66
TEEN	77.26	76.13	72.81	68.16	67.77	64.40	63.25	62.29	61.19	60.32	59.31	66.63
NC-FSCIL	80.45	75.98	72.30	70.28	68.17	65.16	64.43	63.25	60.66	60.01	59.44	67.28
ALFSCIL	79.79	76.53	73.12	69.02	67.62	64.76	63.45	62.32	60.83	60.21	59.30	67.00
DyCR	77.50	74.73	71.69	67.01	66.59	63.43	62.66	61.69	60.57	59.69	58.46	65.82
baseline	74.37	69.98	67.11	63.02	63.04	59.97	59.80	58.25	57.05	56.85	56.26	62.34
baseline+ADBS	79.99	75.89	72.53	68.33	67.92	64.75	64.10	62.93	61.31	60.88	59.65	67.12
	(+5.62)	(+5.91)	(+5.43)	(+5.31)	(+4.88)	(+4.78)	(+4.30)	(+4.69)	(+4.26)	(+4.03)	(+3.39)	(+4.78)
OrCo*	74.58	65.99	64.72	63.06	61.79	59.55	59.21	58.46	56.97	57.99	57.32	61.79
OrCo+ADBS	77.37	71.68	69.62	66.66	64.87	62.71	62.14	61.33	59.78	60.01	58.89	65.01
	(+2.79)	(+5.70)	(+4.90)	(+3.59)	(+3.08)	(+3.17)	(+2.93)	(+2.87)	(+2.80)	(+2.02)	(+1.57)	(+3.22)
ALICE†	77.72	70.19	68.54	65.32	63.78	61.40	60.77	59.82	58.62	58.86	58.51	63.96
ALICE+ADBS	77.97	70.31	68.54	65.37	63.85	61.53	60.92	60.07	58.78	58.97	58.68	64.09
	(+0.24)	(+0.12)	(+0.00)	(+0.05)	(+0.07)	(+0.13)	(+0.15)	(+0.25)	(+0.16)	(+0.11)	(+0.17)	(+0.13)
SAVC*	81.68	77.61	74.84	70.02	69.36	66.61	65.82	64.80	63.14	62.55	62.00	68.95
SAVC+ADBS	81.88	77.86	75.15	70.67	70.49	67.67	67.06	66.03	64.50	63.97	63.32	69.87
	(+0.21)	(+0.25)	(+0.31)	(+0.65)	(+1.13)	(+1.06)	(+1.23)	(+1.24)	(+1.36)	(+1.42)	(+1.32)	(+0.93)

Figure 7: Example of table structure corruption.

Error Type:
Table Structure Corruption

Original Latex

Method	Onl.	QA	PG	Tasks				
				Retr.	TR	AR	AP	PP
ClipBERT	✗	✗	✗	✗	✓	✓	✗	✗
ProcedureVRL	✗	✗	✗	✗	✓	✓	✓	✗
TimeSformer	✗	✗	✗	✗	✓	✓	✓	✓
DistantSup	✗	✗	✗	✗	✓	✓	✓	✓
VideoTF	✗	✗	✗	✗	✓	✓	✓	✓
Paprika	✗	✗	✓	✗	✓	✓	✓	✗
TaskGraph	✗	✗	✓	✗	✓	✓	✓	✗
VideoLLM-onl.	✓	✓	✗	✗	✓	✓	✓	✓
InsTALL (Ours)	✓	✓	✓	✓	✓	✓	✓	✓

Output Latex

Column Number Error

Multirow Error

```
\begin{table}
\centering
\begin{tabular}{lccccccc}
Method & Onl. & QA & PG & \multicolumn{5}{c}{Tasks (TR, AR, AP, PP)} \\
ClipBERT\cite{87} & ✗ & ✗ & ✗ & ✗ & ✓ & ✓ & ✗ & ✗ \\
ProcedureVL\cite{87} & ✗ & ✗ & ✗ & ✗ & ✓ & ✓ & ✓ & ✗ \\
TimeStformerP\cite{7} & ✗ & ✗ & ✗ & ✗ & ✓ & ✓ & ✓ & ✓ \\
DistantSup\cite{44} & ✗ & ✗ & ✗ & ✗ & ✓ & ✓ & ✓ & ✓ \\
VideoT\cite{53} & ✗ & ✗ & ✗ & ✗ & ✓ & ✓ & ✓ & ✓ \\
Paprika\cite{58} & ✗ & ✗ & ✓ & ✗ & ✓ & ✓ & ✓ & ✗ \\
TaskGraph\cite{79} & ✗ & ✗ & ✓ & ✗ & ✓ & ✓ & ✓ & ✗ \\
VideoLLM-onl.\cite{14} & ✓ & ✓ & ✗ & ✗ & ✓ & ✓ & ✓ & ✓ \\
InsTALL (Ours) & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ & ✓ \\
\end{tabular}
\caption{Our work considers an online (Onl.), conversational (QA) setting for procedural tasks, where we can leverage Procedural Graphs (PG). }
\label{table_1}
\end{table}
```

Figure 8: Example of table structure corruption.

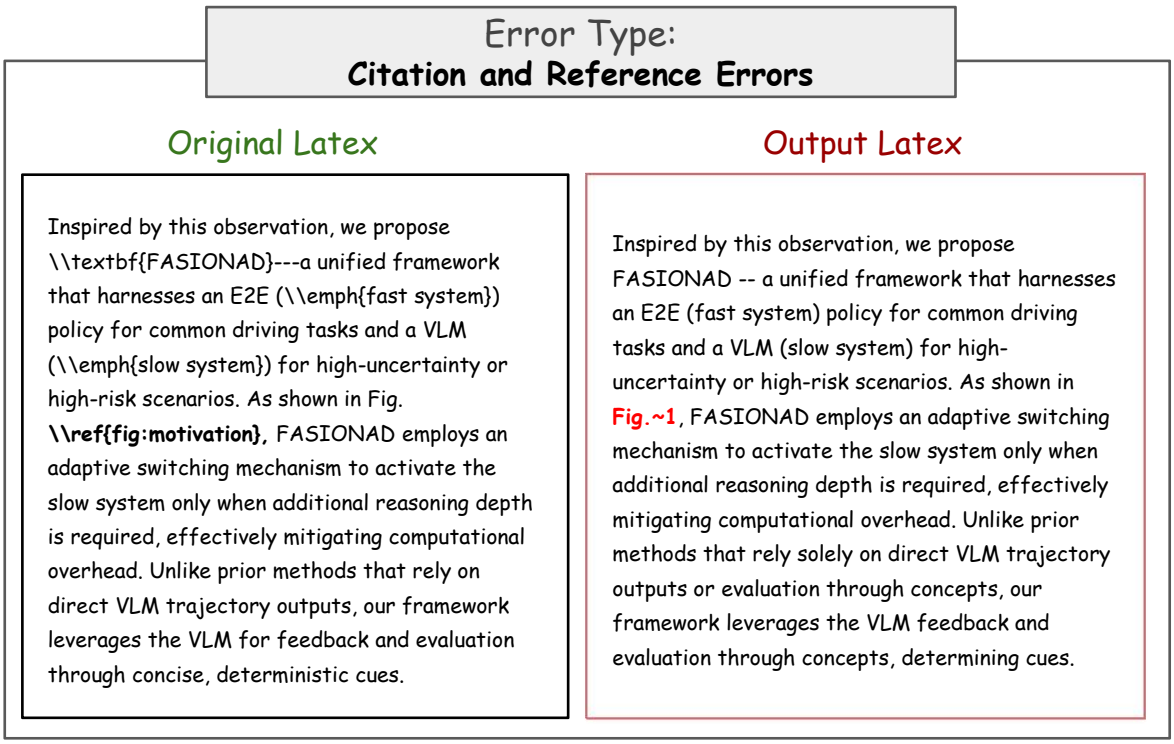

Figure 9: Example of citation and reference errors.

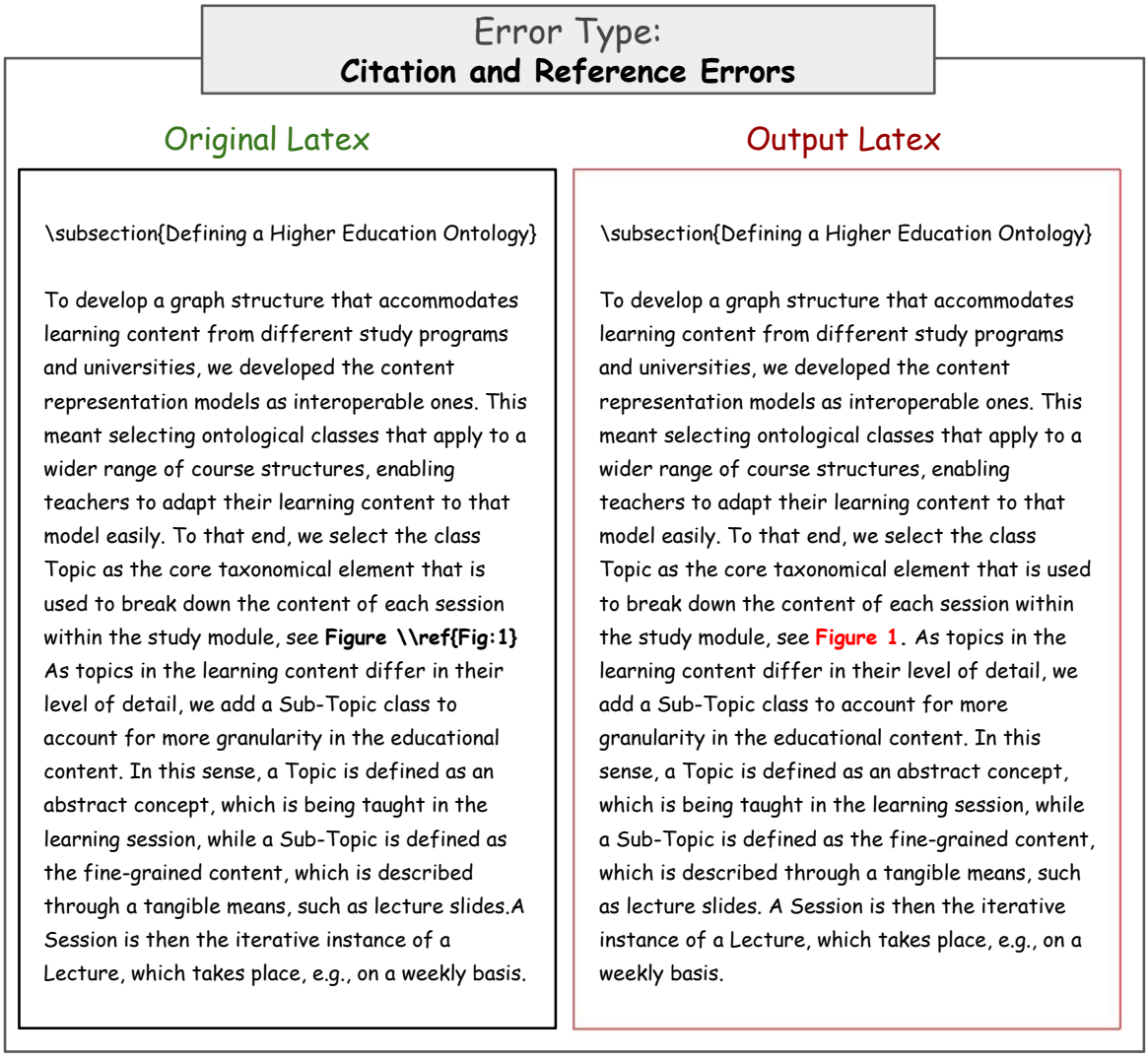

Figure 10: Example of citation and reference errors.

Error Type:
Compilation Failures

Original Latex

Two schemes are used for performance evaluation: a) Direct attacks: Attack the models trained without noise. b) Post-defense attacks: Attack models that have undergone additional defensive training with PGD (the number of iteration $N=50$) for **$5 \\times 10^6$** training steps, keeping other parameters unchanged. This setup demonstrates DAPGD's effectiveness on both benign DNNs (without adversarial training) and robust DNNs (with adversarial training).
The agents performance under various attack methods is measured by the average episode reward obtained over 1000 episodes in three independent experiments. Attacks are applied to the agentsobservation, perturbing the input before it is processed by the policy network. We record the mean reward obtained\nby the agents under each attack scenario, with lower rewards indicating more successful attacks.

Output Latex

Two schemes are used for performance evaluation: a) Direct attacks: Attack the models trained without noise. b) Post-defense attacks: Attack models that have undergone additional defensive training with PGD (the number

Invalid in LaTeX

of iteration N = 50) for 5 10^6 training steps, keeping other parameters unchanged. This setup demonstrates DAPGDs effectiveness on both benign DNNs (without adversarial training) and robust DNNs (with adversarial training).
The agents performance under various attack methods is measured by the average episodic reward obtained over 1000 episodes in 5 independent experiments. Attacks are applied to the agents observation, perturbing the input before it is processed by the policy network. We record the mean reward obtained by the agents under each attack scenario, with lower rewards indicating more successful attacks.

Figure 11: Example of compilation failures.

Error Type:
Compilation Failures

Original Latex

AI can play a moderator role to make human feedback more constructive with a positive tone.

\\textbf{Implication **\\#3**:} To capitalize on the potential of AI tools in SE processes, future AI tools should incorporate personalization features that align feedback delivery with individual preferences. In addition, AI can play the role of a moderator, adjusting or enhancing the tone and constructiveness of feedback to reduce the emotional cost and foster developer well-being.

Output Latex

AI can play a moderate role to make human feedback more constructive with a positive tone.

Invalid in LaTeX

Implication **#3:** To capitalize on the potential of AI tools in SE processes, future AI tools should incorporate personalization features that align feedback delivery with individual preferences. In addition, AI can play the role of a moderator, adjusting or enhancing the tone and constructiveness of feedback to reduce the emotional cost and foster developer well-being.

Figure 12: Example of compilation failures.