TeXformer: Glyphstream-to-LaTeX Transduction for Born-Digital Scientific PDF Reconstruction

Anonymous ACL submission

Abstract

Scientific and technical documents contain dense mathematical and structural information, but much of this content is distributed in formats that expose little usable structure to downstream systems. Existing neural transcription systems often take a redundant route for born-digital documents: they rasterize pages and then recover markup from images, ignoring the fragmentary structured signal already present in the document content stream. We study whether this signal can be used directly. TeXformer converts PDF glyphstreams into normalized LaTeX, using task-specific tokenizers and a normalization-first alignment pipeline. We construct 1.68M aligned local glyphstream–LaTeX examples from arXiv PDF/source pairs, totaling 4.07B task-specific tokens before filtering. On a 2,574-paper held-out benchmark, TeXformer-576M outperforms Nougat on overall, mathematical, structural, math token F1, and BLEU scores, while Nougat remains stronger on plain text transcription. The results show that glyphstream-to-LaTeX conversion is a viable alternative to OCR-style reconstruction for born-digital scientific PDFs, especially when symbolic fidelity matters.

1 Introduction

Scientific natural language processing (NLP), information extraction, and retrieval systems often begin from Portable Document Format (PDF) files rather than author source. PDF is excellent for visual exchange, but poor for computation: mathematical notation, tables, cross-references, and typography carry meaning that is not exposed as ordinary text.

Recent neural systems for scientific document transcription, most prominently Nougat (Blecher et al., 2023), formulate the problem as image-to-markup generation: pages are rasterized and an OCR-like model predicts markup. This generalizes to scanned pages, but discards information that many born-digital PDFs already contain: ordered drawing instructions, glyph identities, font names, and font sizes. These traces are not source code, but they preserve symbolic information that is especially valuable for mathematics.

We ask whether this born-digital representation can be used directly for scientific PDF reconstruction. We call the extracted representation a *glyphstream*: a linear sequence of text spans, font tags, page/paragraph boundaries, and inferred script tags such as `[SUB]` and `[SUP]`. A sentence containing α, δ, and c_n becomes a tagged stream in which text is interleaved with font changes, script boundaries, and hexadecimal codes for non-ASCII mathematical symbols. This is weaker than original LaTeX source, but much stronger than a raster image for symbolic recovery. Table 1 shows the rendered sentence, extracted glyphstream, and LaTeX target.

This paper introduces TeXformer, a glyphstream-to-LaTeX model for born-digital scientific PDFs. TeXformer is trained from scratch as an encoder–decoder transformer with task-specific source and target tokenizers. Its dataset pipeline pairs arXiv PDFs with source bundles, normalizes sources before chunking, and aligns each full document into local training pairs matching glyphstream spans to normalized LaTeX spans for the same document region. The final setup uses 1.23M filtered examples from 103,327 papers, starting from a pre-filter train split of 3.86B tokens; the held-out benchmark evaluates 2,574 papers. The anonymous code repository is available at https://anonymous.4open.science/r/TeXformer-EMNLP-2026-48CB; the codebase and trained models will be released publicly after review. The dataset is also intended for public release once distribution and licensing questions around source-derived data have been resolved.

The central finding is that representation and supervision quality matter more than treating PDF reconstruction as ordinary OCR. TeXformer-576M reaches 0.719 overall score on the held-out bench-

Rendered PDF	The rest of parameters $\alpha(0), \delta, c_n, s_n$ $(n = 1, 2, x)$ are fixed from trajectory fit
Glyphstream	`The rest of parameters[CMMI10] <03B1>[CMR10](0)[CMMI10] <002C> <03B4> <002C> c[SUB][CMMI7]n[/SUB] [CMMI10]<002C> s[SUB][CMMI7]n[/SUB] [CMR10] ([CMMI10]n[CMR10] = 1[CMMI10]<002C>[CMR10] 2[CMMI10]<002C> x[CMR10]) are fixed from trajectory fit`
LaTeX source	`The rest of parameters $\alpha(0), \delta , c_n, s_n$ ($n = 1, 2, x$) are fixed from trajectory fit`

Table 1: Example of the three representations involved in the task: the rendered PDF sentence, the extracted glyphstream, and the normalized LaTeX target. Font switches are retained explicitly, and subscript boundaries are marked in the glyphstream rather than inferred from a raster image.

mark, compared with 0.674 for Nougat-base, with larger gains on mathematics and structure. Nougat remains stronger on text-only transcription, but glyphstream transduction is consistently competitive and often preferable for math-heavy, structure-sensitive papers.

We organize the paper around three research questions (RQs): whether glyphstreams can recover symbolic and mathematical content from born-digital PDFs; how sensitive reconstruction is to alignment and filtering decisions; and how glyphstream reconstruction compares with image-based baselines such as Nougat. We find that glyphstreams support semantically faithful LaTeX recovery, dataset construction is a central determinant of performance, and TeXformer is stronger on overall, mathematical, and structural recovery while Nougat remains stronger on text-only transcription.

Our contributions are:

- For **RQ1**, we define a born-digital scientific PDF reconstruction task that maps linearized PDF glyphstreams to semantically faithful normalized LaTeX without rasterization.
- For **RQ2**, we contribute a large glyphstream–LaTeX dataset from arXiv PDF/source pairs, a normalization-first full-document alignment and filtering pipeline, and task-specific source and target tokenizers for constructing reliable local training pairs.
- For **RQ3**, we introduce TeXformer, a family of from-scratch encoder–decoder transformers for this task, including 100M, 270M, and 576M parameter checkpoints, and evaluate them against Nougat on a 2,574-paper held-out benchmark.

2 Related Work

Scientific PDF reconstruction connects document understanding, scholarly text processing, mathematical OCR, and long-context generation. We organize prior work by representation choice and problem setting.

Scientific PDF processing. Scientific PDF processing has long been studied as document analysis and digital library infrastructure. Systems such as GROBID (Lopez, 2009) and CERMINE (Tkaczyk et al., 2015) extract bibliographic metadata, body text, and structure from scholarly PDFs. Later datasets and models for layout and structured extraction include DocBank (Li et al., 2020), LayoutLM (Xu et al., 2020), DocLayNet (Pfitzmann et al., 2022), and VILA (Shen et al., 2022). These systems recover document structure, but their primary output is not normalized LaTeX with mathematical notation.

Within NLP, corpus-building efforts such as S2ORC (Lo et al., 2020) enabled large-scale scholarly text mining once papers were converted into machine-usable formats. We address the earlier bottleneck of recovering symbolically faithful markup from born-digital PDFs when source files are unavailable at inference time.

PDF-to-markup and mathematical transcription. Image-to-markup models for mathematical expressions and documents typically operate on rendered images (Deng et al., 2017). Prior PDF-to-LaTeX work has considered neural conversion of mathematical PDFs (Wang and Liu, 2020), and OCR-free models such as Donut (Kim et al., 2022) show that end-to-end document transduction can work without an explicit OCR stage. Nougat (Blecher et al., 2023) is the strongest modern baseline for academic document transcription and predicts markup from rasterized pages. This is general,

but mismatched to born-digital PDFs, whose glyph identities, font switches, and superscript/subscript cues are partly discarded by rasterization.

Our setting therefore differs in both input representation and output objective. TeXformer uses born-digital glyphstreams instead of images, and it is evaluated at the LaTeX level with particular attention to mathematical and structural fidelity. The goal is not generic document OCR, but faithful recovery of machine-usable scientific notation from documents that already expose it internally.

Datasets and supervision for document transduction. The dataset story matters as much as the model story. Prior document datasets often emphasize layout regions, OCR text, or downstream annotations rather than aligned source-to-document transduction pairs. By contrast, TeXformer requires local pairs between glyphstream spans and normalized LaTeX spans. This makes alignment quality a first-order issue rather than a preprocessing detail. Earlier approaches based on weak alignment, including SyncTeX-style heuristics, are useful for bootstrapping, but too noisy for stable long-form generation when local boundaries drift. Our main dataset contribution is therefore not only scale, but a normalization-first, PDF-aware alignment pipeline designed specifically for glyphstream-to-LaTeX supervision.

Long-context sequence modeling. Scientific papers are long documents, so PDF reconstruction is also a long-form generation problem. The Text-to-Text Transfer Transformer (T5) (Raffel et al., 2020), Longformer Encoder-Decoder (LED) (Beltagy et al., 2020), and LongT5 (Guo et al., 2022) provide natural baselines or architectural reference points for long sequence-to-sequence modeling. In our preliminary experiments, directly adapting pretrained text models was less effective than building a task-specific representation and tokenizer pair. TeXformer therefore keeps the encoder–decoder formulation (Vaswani et al., 2017), but trains from scratch on the glyphstream-to-LaTeX task.

Task-specific tokenization. Task-specific tokenization is common when generic tokenizers fragment meaningful units, as in rare-word subword modeling (Sennrich et al., 2016) or SentencePiece-style training (Kudo and Richardson, 2018). Our case is more extreme: the source has bracketed PDF tags, font identifiers, and hexadecimal glyph codes, while the target has LaTeX control words and math-heavy syntax. TeXformer therefore learns separate source and target vocabularies on the aligned corpus.

3 Task Definition

Given a born-digital scientific PDF P, the goal is to produce normalized LaTeX Y that preserves textual, mathematical, and tabular content well enough for downstream information extraction and retrieval. We do not attempt source-code decompilation: author macros, package structure, TikZ programs, and exact formatting are outside scope. We target semantically faithful recovery of scientifically meaningful content in compilable markup.

The input to TeXformer is a glyphstream $G = (g_1, \ldots, g_n)$. Each glyphstream token may be ordinary text, a font tag, a structural boundary, or a script marker. The representation is deliberately restricted to information recoverable from the PDF content stream. Scanned documents are outside the scope, and figures are not reconstructed because the glyphstream does not contain the original bitmap or vector figure asset.

The target is normalized LaTeX. During dataset construction, source bundles are converted into document bodies with local macros expanded, references normalized, bibliography material removed, and figures replaced by placeholders when captions are available. Tables are retained in the final targets.

4 Dataset Construction

The dataset is a central contribution. TeXformer needs large-scale, boundary-faithful supervision: glyphstream spans and LaTeX targets must correspond closely enough for structured generation to be learnable. The pipeline therefore collects a mathematics-heavy born-digital corpus, extracts symbolic PDF signal, normalizes source bundles, and aligns each full document into local glyphstream-span/LaTeX-span pairs before tokenization and filtering.

4.1 Corpus Acquisition

We build the dataset from arXiv papers (Clement et al., 2019) for which both a PDF and source bundle are available. The scrape focuses on mathematics-heavy categories from 2018 onward, including mathematics, theory-heavy physics, computer science, statistics, and signal processing. The raw collection contains 118,535 paper directories

occupying approximately 199 gigabytes (GB) of PDF-plus-source storage. Of these, 112,715 contain both PDF and LaTeX source, corresponding to 95.1% valid PDF–source pairs.

The corpus is intentionally symbolic. Mathematics is the largest primary category with 62,016 papers, followed by computer science with 17,980. Across 116,323 PDFs with measured page counts, the collection contains 2.82M pages, with a mean of 24.3 and median of 19 pages per paper. This scale motivates local aligned examples and hierarchical inference.

4.2 Glyphstream Extraction

The PDF side is extracted without rasterization. We parse page content streams, remove common noise such as running headers, page numbers, footer artifacts, and arXiv stamps, detect multi-column reading order, and serialize the remaining spans into a tagged linear sequence. The extractor emits structural tags such as `[PAGE]`, `[COL]`, `[PARA]`, and `[BR]`. Font identity is retained through tags such as `[CMR10]` and `[CMMI10]`. Super- and subscript regions are inferred from geometry and marked with `[SUP]`, `[/SUP]`, `[SUB]`, and `[/SUB]`.

Character normalization is font-aware. Ordinary text fonts are normalized into readable text with whitespace cleanup and ligature expansion. Math-italic fonts preserve American Standard Code for Information Interchange (ASCII) alphanumerics directly, but emit other symbols as hexadecimal codes.

4.3 LaTeX Normalization

On the target side, source bundles are normalized before alignment or chunking. The pipeline identifies the main `.tex` file, recursively inlines `\input` and `\include` files, extracts the document body, and expands user-defined macros when possible. It removes hidden metadata, strips bibliography material, normalizes labels and references, rewrites citations against rendered order, and replaces figures by placeholders while preserving captions. The final targets retain tabular material because tables can often be reconstructed once supervision noise is controlled.

4.4 Full-Document Alignment

The main dataset challenge is producing local pairs whose boundaries actually match. Full papers exceed practical video random-access memory (VRAM) budgets because encoder/decoder attention and activation memory grow sharply with sequence length, forcing chunking during training and inference. Once chunking is unavoidable, the model needs many local glyphstream–LaTeX pairs whose boundaries correspond closely enough for structured generation to be learnable.

Chunking alone does not solve this. PDF rendering order and LaTeX source order are only approximately monotonic: front matter, packages, macros, and bibliography material may not render, while floating figures and tables may be declared far from where they appear. Small offset errors can pair a glyphstream chunk with a target that omits an equation or still contains non-rendered source, giving the model contradictory supervision.

We therefore align at the full-document level after normalization. The aligner finds sentence-level anchor matches between the glyphstream and normalized LaTeX stream, constructs a monotonic alignment chain, repairs local boundaries, and scores candidate pairs with a combined metric:

$$s_{\text{align}} = 0.25F_1^{\text{content}} + 0.30s_{\text{seq}} + 0.25s_{\text{mono}} + 0.15r_{\text{pos}} + 0.05s_{\text{len}}. \tag{1}$$

Here F_1^{content} is frequency-weighted content-word overlap after markup stripping; s_{seq} is weighted 2-, 3-, and 4-gram overlap; s_{mono} measures monotonicity; r_{pos} is non-negative Pearson correlation between matched positions; and s_{len} penalizes source–target length mismatch:

$$s_{\text{len}} = \frac{1}{1 + |\log(n_{\text{pdf}}/n_{\text{tex}})|}. \tag{2}$$

Chunk formation respects LaTeX structure where possible, preferring boundaries that avoid splitting proofs, lists, and theorem-like blocks. The document-level chunker remains imperfect: very long contexts and deeply nested environments can force boundary compromises. The first alignment pass uses a permissive cutoff of 0.55, yielding 111,974 processed papers and 1,389,946 aligned samples from 112,715 valid PDF–source pairs.

4.5 Task-Specific Tokenization

We train separate 48k-token byte-level byte-pair encoding (BPE) tokenizers for source glyphstreams and target LaTeX using the first-pass aligned corpus. Frequent PDF tags and LaTeX commands are isolated before BPE training and promoted to dedicated vocabulary items, giving the model atomic units such as `[CMMI10]` and `\alpha`.

This improves context efficiency because over-fragmenting font tags, script markers, encoded symbols, control sequences, and environments wastes encoder and decoder budget. Table 2 shows a representative example.

Table 3 summarizes the main tokenized dataset before training-time filtering: 1.68M examples and 4.07B tokens overall, including 1.60M train examples and 3.86B train tokens.

The final dataloader removes noisy alignments using thresholds over alignment score, recall, content overlap, and source–target length ratio. These thresholds retain 76.6% of the raw training split, reducing it from 1,598,885 to 1,225,141 examples.

5 Model

With the dataset and tokenizer pair fixed, we keep the architecture conservative to test whether a task-specific encoder–decoder transformer can exploit glyphstreams once supervision is reliable.

TeXformer is trained from scratch. The encoder consumes glyphstream tokens, and the decoder autoregressively generates normalized LaTeX. The architecture follows the standard sequence-to-sequence transformer (Vaswani et al., 2017), with root mean square normalization (RMSNorm) (Zhang and Sennrich, 2019), rotary position embeddings (RoPE) (Su et al., 2024), Swish-gated linear unit (SwiGLU) feed-forward blocks (Shazeer, 2020), tied decoder input/output embeddings, and key-value (KV) caching during generation.

Table 4 lists the released model family. Variants differ in width, feed-forward dimension, depth, and attention heads, while preserving a symmetric encoder/decoder template.

6 Training and Inference

Training and inference share the same constraint: scientific papers are too long and heterogeneous to treat as a single flat sequence, even with a symbolic representation. We therefore combine local training examples with hierarchical document-level decoding.

All final models use AdamW, label smoothing $\epsilon = 0.1$, linear warmup, cosine annealing (Loshchilov and Hutter, 2016), bfloat16 arithmetic, and distributed data parallel training. The 576M model also uses gradient checkpointing. The published runs use two RTX 5090 graphics processing units (GPUs).

Document-level inference uses a hierarchical chunker: page boundaries, paragraph boundaries, weaker sentence-like boundaries, and finally hard character splits when no natural boundary satisfies the token budget. Generated chunks are wrapped into a compilable article template. This hierarchy is still imperfect on long contexts and nested environments. The default decoding uses temperature 0.6, top-$p = 0.95$, top-$k = 12$, an n-gram penalty of 1.2 with $n = 12$, and a maximum generation length of 2,560 tokens.

7 Evaluation

We evaluate checkpoint quality and full-paper reconstruction. Because the claim is symbolically faithful recovery rather than plain-text OCR, text quality alone is insufficient and mathematical fidelity must be measured directly.

7.1 Metrics

We report a composite score over text, mathematical notation, and structure. Text combines word accuracy, character n-gram F-score (chrF), and $1-$ word error rate (WER); math combines math token F1, math-structure similarity, math chrF, and $1-$ character error rate (CER); structure combines LaTeX validity, environment F1, and command F1. The overall score weights math more for math-heavy samples and text more for text-only samples. We also report math token F1 and Bilingual Evaluation Understudy (BLEU).

7.2 Baseline

We compare against Nougat-small and Nougat-base (Blecher et al., 2023). Because Nougat emits Markdown, we convert its output into LaTeX before scoring. As a cross-format check, we also converted TeXformer outputs from LaTeX to Markdown and observed the same qualitative ordering: TeXformer remained stronger on math- and structure-sensitive content, while Nougat remained stronger on text-centric transcription. We report the LaTeX-space benchmark because LaTeX is TeXformer's target.

8 Results

The results emphasize two claims: TeXformer depends on dataset construction, especially alignment and tokenization, and the resulting model family is competitive with Nougat while stronger on mathematically sensitive content.

Tokenizer	Encoder fragment tokenization of `[CMMI10]` `<03B1>[CMR10]`	Count	Decoder tokenization of `\alpha`	Count
T5 / LongT5	`[` `CM` `MI` `10` `]` `␣` `<` `03` `B` `1` `>` `[` `C` `MR` `10` `]`	16	`\` `al` `pha`	3
LED	`[` `CM` `MI` `10` `]` `␣<` `03` `B` `1` `>` `[` `C` `MR` `10` `]`	15	`\` `alpha`	2
TeXformer	`[CMMI10]` `␣<` `03` `B` `1` `>` `[CMR10]`	**7**	`\alpha`	**1**

Table 2: Why TeXformer uses task-specific tokenizers. Generic text tokenizers fragment PDF font tags and LaTeX commands, spending context on representation rather than document content.

Split	Samples	Input tokens	Output tokens	Total tokens	Avg. input	Avg. output
Train	1,598,885	2.271B	1.587B	3.858B	1,421	993
Validation	42,729	60.7M	42.5M	0.103B	1,421	994
Test	43,229	61.1M	42.8M	0.103B	1,414	989
Total	**1,684,843**	**2.393B**	**1.672B**	**4.065B**	**1,421**	**993**

Table 3: Corpus-level token statistics under TeXformer's task-specific tokenizers, before the final training-time quality filter.

Variant	d_{model}	d_{ff}	Enc.	Dec.	Params
100M	512	2048	6	6	96.4M
270M	768	3072	12	12	272.0M
576M	**1024**	**4096**	**16**	**16**	**576.5M**

Table 4: Released TeXformer model family.

Model	Overall	Text	Math	Struct	Math F1
100M	0.8631	0.9394	0.8096	0.8243	0.8993
270M	0.8701	0.9415	0.8194	0.8351	0.9040
576M	**0.8717**	**0.9424**	**0.8215**	**0.8366**	**0.9048**

Table 5: Validation results for TeXformer checkpoints trained on the final aligned dataset.

8.1 Alignment and Dataset Quality

An earlier SyncTeX-style alignment route was explored, but abandoned because local boundaries were too noisy for stable generation. The decisive change was rebuilding supervision around full-document normalization, PDF-aware alignment, and training-time filtering; final model quality is therefore best understood as a corpus improvement rather than a purely architectural jump.

8.2 Checkpoint Validation

Table 5 reports the best validation checkpoints. Scaling from 100M to 576M improves all metrics, but gains are modest after 100M, suggesting a data-limited regime.

8.3 Held-Out Benchmark Against Nougat

Table 6 shows full-paper results over 2,574 held-out papers. Nougat's Markdown output is converted to LaTeX before scoring against the reference. TeXformer-576M obtains the best overall, math, structure, math token F1, and BLEU scores; Nougat-small obtains the best text score. Thus image-based transcription remains strong for surface text, while glyphstream transduction better preserves mathematical and structural content in born-digital PDFs. Figure 2 in the appendix gives a qualitative formula-heavy example.

8.4 Math-Heavy Single-Paper Diagnostic

Table 7 isolates a strongly mathematical case: a 50-page held-out paper with 1,694 reference math blocks. The TeXformer advantage grows in this setting: even TeXformer-100M outperforms Nougat-base on all metrics, and TeXformer-576M reaches 0.9691 math token F1.

9 Analysis

The results point to three main sources of performance: the glyphstream representation, cleaner supervision, and additional aligned data.

Glyphstreams are useful, but not complete. Born-digital glyphstreams preserve enough symbolic signal to reconstruct substantial portions of scientific papers, including formulae and many tables. They are not source code and do not contain author macros, package structure, or figure assets,

Model	Size	Overall	Text	Math	Struct	Math F1	BLEU
TeXformer-100M	96M	0.7039	0.7007	0.7461	0.6268	0.8724	0.6934
TeXformer-270M	272M	0.7145	0.7208	0.7500	0.6317	0.8712	0.6994
TeXformer-576M	576M	**0.7190**	0.7160	**0.7631**	**0.6377**	**0.8820**	**0.7097**
Nougat-small	250M	0.6758	**0.7916**	0.6580	0.4810	0.7468	0.6548
Nougat-base	350M	0.6737	0.7875	0.6576	0.4791	0.7465	0.6525

Table 6: Aggregate LaTeX-level benchmark over the 2,574-paper held-out test split. Nougat Markdown is converted to LaTeX before scoring; converting TeXformer LaTeX to Markdown gives the same qualitative ordering.

Model	Overall	Text	Math	Struct	Math F1	BLEU
TeXformer-100M	0.7317	0.8075	0.7200	0.6034	0.9300	0.6556
TeXformer-270M	0.7750	0.8778	0.7601	0.5993	0.9594	0.6891
TeXformer-576M	**0.8144**	**0.9259**	**0.7801**	**0.6603**	**0.9691**	**0.7116**
Nougat-small	0.5219	0.7059	0.4006	0.3965	0.5545	0.3363
Nougat-base	0.5230	0.7305	0.3821	0.3894	0.5302	0.3339

Table 7: Single-paper diagnostic on a math-heavy 50-page held-out paper with 1,694 reference math blocks.

but they preserve the low-level traces needed for mathematical recovery.

Supervision quality dominates naive scaling. Noisy boundary supervision can overwhelm model scale. Earlier weakly aligned pipelines, including SyncTeX-style heuristics, were not reliable enough for stable long-form transduction. After normalization-first alignment and quality filtering, scaling from 100M to 576M helped consistently, but with diminishing returns.

The current regime is still data-limited. Figure 1 suggests TeXformer has not saturated the available supervision. This is notable because TeXformer starts from 112,715 valid PDF–source pairs, whereas Nougat reports 1,748,201 training papers (Blecher et al., 2023). Despite this data disadvantage, TeXformer outperforms Nougat on overall, math, and structure metrics; the steady gains from 100M to 270M to 576M suggest that larger or cleaner aligned corpora would likely help further.

TeXformer and Nougat have complementary strengths. Nougat's text score is higher on the aggregate benchmark, while TeXformer is stronger on math and structure. Rasterized images preserve visual shapes and typography, benefiting text transcription; glyphstreams preserve machine-readable symbolic and font information, benefiting mathematical recovery.

10 Conclusion

We introduced TeXformer, a glyphstream-based encoder–decoder model for reconstructing scientific LaTeX from born-digital PDFs. Instead of rasterizing pages, TeXformer uses symbolic traces already present in PDF content streams. A large aligned arXiv corpus, task-specific tokenizers, and normalization-first alignment are central to the result. On a 2,574-paper held-out benchmark, TeXformer outperforms Nougat on overall, mathematical, and structural reconstruction, while remaining weaker on pure text transcription. Glyphstream transduction is therefore a strong representation for math-rich scientific document NLP, where symbolic fidelity matters for retrieval and information extraction.

11 Limitations

TeXformer applies only to born-digital PDFs whose content streams expose machine-readable glyphs and fonts. It is not designed for scanned documents or image-only PDFs. The model also does not reconstruct figures, author-defined macro structure, bibliography source, package choices, or exact source formatting. Its objective is semantically faithful recovery of textual, mathematical, and often tabular scientific content.

Long-form generation also remains difficult. Although hierarchical chunking makes full-paper reconstruction practical, very long generations remain harder than short or medium chunks. Low repetition and high math fidelity are not always optimized by the same decoding hyperparameters, so future work should investigate objectives and decoding strategies that better preserve long-range consistency.

The main external comparison is against Nougat, whose native output is Markdown rather than LaTeX.

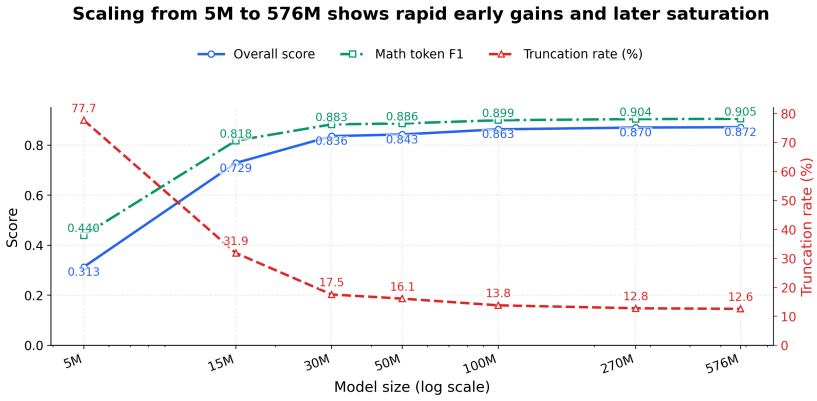

Figure 1: Scaling behavior across TeXformer model sizes. Validation quality improves from 100M to 270M to 576M without clear saturation, supporting the interpretation that aligned glyphstream–LaTeX supervision remains limiting.

We convert Nougat output to LaTeX before scoring, so the comparison is necessarily cross-format. This makes the benchmark conservative in some ways and imperfect in others. More baselines and direct human evaluation of reconstructed documents would strengthen the empirical picture.

Any cross-format pipeline introduces its own normalization effects, so the exact margins should be interpreted more cautiously than the overall ordering.

The corpus is derived from arXiv papers and is intentionally mathematics-heavy. Results may not transfer directly to older PDFs, publisher PDFs with unusual fonts, documents with complex vector graphics, or scientific domains whose notation and layout conventions differ substantially from the scraped categories.

12 Ethical Considerations

The dataset is built from publicly available scientific PDFs and source bundles. The code and trained models will be released publicly for research use. The dataset is also intended for public release, but source availability does not remove all distribution and licensing concerns, so release must follow a review of redistribution conditions, provenance requirements, and any necessary takedown process. Any public dataset release should preserve provenance, document licenses where available, and provide clear instructions for removing examples if required. The system may also produce plausible, but incorrect reconstructions, especially in ordinary prose around otherwise correct mathematical content. Downstream users should not treat generated LaTeX as an authoritative replacement for the original paper.

The positive impact of this work is improved access to scientific content for search, annotation, accessibility, and downstream mathematical NLP. The main risk is silent error propagation: a reconstructed formula or identifier may be used by later systems as if it were ground truth. For this reason, reconstruction outputs should be paired with confidence diagnostics and, where possible, evaluated against original sources.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

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

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. 2019. On the use of arxiv as a dataset. *arXiv preprint arXiv:1905.00075*.

Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, and Alexander M. Rush. 2017. Image-to-markup generation with coarse-to-fine attention. In *International Conference on Machine Learning*, pages 980–989. PMLR.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Won-

seok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *Computer Vision – ECCV 2022*, pages 498–517. Springer.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. 2020. Docbank: A benchmark dataset for document layout analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 949–960.

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.

Patrice Lopez. 2009. GROBID: Combining automatic bibliographic data recognition and term extraction for scholarship publications. In *Research and Advanced Technology for Digital Libraries*, pages 473–474, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ilya Loshchilov and Frank Hutter. 2016. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3743–3751.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Noam Shazeer. 2020. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Zejiang Shen, Kyle Lo, Lucy Lu Wang, Bailey Kuehl, Daniel S. Weld, and Doug Downey. 2022. VILA: Improving structured content extraction from scientific PDFs using visual layout groups. *Transactions of the Association for Computational Linguistics*, 10:376–392.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Dominika Tkaczyk, Pawel Szostek, Mateusz Fedoryszak, Piotr Jan Dendek, and Lukasz Bolikowski. 2015. Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition*, 18(4):317–335.

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

Zelun Wang and Jyh-Charn Liu. 2020. Pdf2latex: A deep learning system to convert mathematical documents from pdf to latex. In *Proceedings of the ACM Symposium on Document Engineering 2020*, pages 1–10.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1192–1200.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *arXiv preprint arXiv:1910.07467*.

A Additional Reproducibility Notes

The dataset construction pipeline proceeds in the following order: raw arXiv PDF/source acquisition, glyphstream extraction, LaTeX source normalization, full-document alignment, tokenizer training, final dataset generation, training-time quality filtering, model training, and full-paper inference. The ordering matters: tokenizers are trained after first-pass alignment, and the final training corpus is regenerated only after the task-specific tokenizer pair exists.

B Additional Dataset Details

The final dataset covers 103,327 papers at the corpus level and has an average alignment score of 0.8603 and average content F1 of 0.9164. After the hybrid filter, the final training split contains 1,225,141 examples. The unfiltered validation and test splits contain 42,729 and 43,229 examples, respectively.

C Additional Decoding Details

The decoding sweep considered temperatures $\{0.1, 0.3, 0.5, 0.6, 0.7, 0.8, 0.9\}$, n-gram penalties $\{1.0, 1.1, 1.2, 1.3, 1.4, 1.5\}$, and n-gram sizes $\{2, 4, 6, 8, 10, 12, 14\}$. The selected default setting was temperature 0.6 with an n-gram penalty of 1.2 and $n = 12$, which gave the best aggregate operating point among representative full-paper decoding settings.

D Qualitative Nougat Comparison

Figure 2 shows a representative formula-heavy comparison between the reference, Nougat after Markdown-to-LaTeX conversion, and TeXformer output. The example illustrates the same pattern as the aggregate benchmark: Nougat remains fluent on surrounding text, while TeXformer better preserves the mathematical expression.

Ground Truth

Now, let $\vec{G}^{(l)}$ be the directed acyclic hypergraph obtained from G and using the deletion ordering $v_0, ..., v_{n-1}$. We will have the following.

$$d^{+}_{\vec{G}^{(l)},l}(v_i) \leq d_{G\langle\{v_i,...,v_n\}\rangle_l}(v_i) \cdot r(G) \leq \kappa_l(G) \cdot r(G) \tag{1}$$

The first inequality holds from the fact that the l-outdegree of the vertex v_i will be bounded by its degree in the induced l-trimmed subhypergraph at the time of deletion times the rank of the hypergraph. The second inequality holds because v_i is a vertex with minimum degree in $G\langle\{v_i, ..., v_n\}\rangle_l$, and by definition the minimum degree of any induced l-trimmed subhypergraph is bounded by $\kappa_l(G)$. Finally, we will have:

nougat-0.1.0-base

Now, let $\vec{G}^{(l)}$ be the directed acyclic hypergraph obtained from G and using the deletion ordering $v_0, ..., v_{n-1}$. We will have the following.

$$d^{+}_{\vec{G}^{(l)},l}(v_i) \leq d_{G(\{v_i,...,v_n\})}(v_i) \cdot r(G) \leq \kappa_l(G) \cdot r(G) \tag{1}$$

The first inequality holds from the fact that the l-outdegree of the vertex v_i will be bounded by its degree in the induced l-trimmed subhypergraph at the time of deletion times the rank of the hypergraph. The second inequality holds because v_i is a vertex with minimum degree in $G\langle\{v_i, ..., v_n\}\rangle_l$, and by definition the minimum degree of any induced l-trimmed subhypergraph is bounded by $\kappa_l(G)$. Finally, we will have:

TeXformer-100M

Now, let $\vec{G}^{(l)}$ be the directed acyclic hypergraph obtained from G and using the deletion ordering $v_0, ..., v_{n-1}$. We will have the following.

$$d^{+}_{\vec{G}^{(l)},l}(v_i) \leq d_{G\langle\{v_i,...,v_n\}\rangle_l}(v_i) \cdot r(G) \leq \kappa_l(G) \cdot r(G) \tag{1}$$

The first inequality holds from the fact that the l-outdegree of the vertex v_i will be bounded by its degree in the induced l-trimmed subhypergraph at the time of deletion times the rank of the hypergraph. The second inequality holds because v_i is a vertex with minimum degree in $G\langle\{v_i, ..., v_n\}\rangle_l$, and by definition the minimum degree of any induced l-trimmed subhypergraph is bounded by $\kappa_l(G)$. Finally, we will have:

Figure 2: Qualitative comparison between the ground-truth rendered passage, Nougat-base output after Markdown-to-LaTeX conversion and recompilation, and TeXformer output. The highlighted region shows a mathematical transcription error in Nougat, where angle brackets are rendered as parentheses, while TeXformer preserves the formula.