arXiv:2512.18115v1 [cs.MM] 19 Dec 2025

用于将学术 PDF 文件高效转换为 Markdown 的布局感知文本编辑

Changxu Duan[0000−0003−0547−0901]

达姆施塔特工业大学，达姆施塔特，德国
duan@linglit.tu-darmstadt.de

摘要。 以 PDF 格式存储的学术文档可以转换为纯文本结构化标记语言，从而提升可访问性，并支持可扩展的数字图书馆工作流程。标记语言更便于更新和定制，使学术内容能够更好地适应多样化的使用场景，例如语言语料库编纂。此类通常以 PDF 格式提供的文档包含数学公式、图、页眉和表格等复杂元素，以及布局密集的文本。现有的端到端解码器 Transformer 模型能够将文档屏幕截图转换为标记语言。然而，这些模型效率低下：它们从头开始进行逐词元解码，在重新生成本可直接从 PDF 文件复制的密集文本时浪费了大量推理步数。为解决这一问题，我们提出了 EditTrans，这是一种混合编辑生成模型，其特征能够在开始生成标记语言之前，识别 PDF 中待编辑文本的序列。EditTrans 包含一个轻量级分类器，该分类器基于文档版面分析模型进行微调，训练数据来自 arXiv 中的 162,127 页文档。在评估中，与端到端解码器 Transformer 模型相比，EditTrans 将转换延迟最多降低了 44.5%，同时保持了转换质量。我们的代码和可复现的数据集生成脚本均已开源。

关键词： PDF 到 Markdown 转换 · 文档布局分析 · 文本编辑模型 · 中间填充。

1 引言

将学术文档从 PDF 转换为 HTML 或 Markdown 等标记语言，可显著提升其可访问性和易用性。这种转换不仅能改善网络无障碍，还能增强文档交互性、提高可搜索性和索引效率，并确保内容兼容不同平台 [8]。这对于大规模数字图书馆项目尤为重要，也能通过使内容可供机器处理来支持 FAIR 原则 [43]。此类文档通常以 PDF 格式交付，包含数学公式、图、页眉和表格等复杂元素，以及布局密集的文本，给计算文档处理带来了重大挑战 [20]。

文档理解涉及对排版丰富的文档进行自动分类和信息提取 [5]。尽管基于 Transformer 的模型 [12] 和近期的端到端解码器 [4, 22, 33] 能够将 PDF 文件转换为 Markdown，但它们存在一个核心效率问题：必须从头开始逐个生成文本词元，导致推理速度缓慢并产生不必要的计算开销。对于文本中很大一部分本可直接复制的高密度学术文档，这一问题尤为突出。

尽管效果显著，基于解码器的模型在处理速度方面仍存在不足。它们从头开始逐个生成文本词元，导致推理耗时较长，尤其是在文本密集的学术文档中，因为其中大部分文本都可以直接从 PDF 文件复制。这种序列解码方式带来了不必要的计算开销，使转换过程明显变慢。

为缓解这些不足，我们从两种互补范式中汲取灵感。首先，相较于从头生成，基于编辑的方法更适用于语法错误纠正等任务 [25], 。与序列到序列模型不同，基于编辑的模型只识别并修改文档中必要的部分，从而显著提升推理速度。其次，为了处理公式和表格等复杂且无法复制的元素的生成，我们采用中间填充（FIM）技术 [3]。FIM 擅长在同时以之前和之后的上下文为条件的情况下插入内容，因此非常适合我们的混合方法。

为了解决现有模型效率低下的问题，我们提出了 EditTrans。虽然我们的方法可以视为一种工作流优化，但其研究创新在于有效融合布局感知分类、文本编辑原则和中间填充生成，直接针对最先进文档 Transformer 的冗余瓶颈。EditTrans 采用轻量级分类器，该分类器基于 DLA 模型，在来自 arXiv 的 162,127 页文档上进行微调，用于构建结构化编辑队列，并仅在必要时优先进行修改。本文的贡献如下：

– **EditTrans：**一种混合编辑生成模型，可在启动标记生成之前，预先识别并排队等待编辑的 PDF 文本。该方法显著减少了冗余文本生成，并提高了效率。– 为增强可复现性和透明度，我们公开了数据集制作脚本，以及实验中使用的 arXiv 文档标识符。我们的开源代码可在线获取。[1]– 我们改进了 PDF 到标记的评测数据集，为比较旨在提升学术文档可访问性的工具提供了更公平、更可靠的基准。

[1] https://github.com/Fireblossom/EditTrans

2 相关工作

2.1 学术文档转换

GROBID [14] 是一个机器学习库，可用于提取、解析和重构文档，包括将 PDF 转换为编码为结构化 XML 的文档。然而，它不够灵活，因为它会将公式和表格转换为图像，从而妨碍后续访问。docTR [27] 和 DocBed [51] 会先识别文档布局，再提取文本内容。Donut [15] 是一种文档理解模型，由视觉编码器和语言模型解码器组成，无需直接从文档中获取文本。Nougat [4] 延续了 Donut 的做法，实现了学术文档的截图到 Markdown 转换。LOCR [39] 使用额外的位置提示，解决了 Nougat 的幻觉和重复问题。Kosmos-2.5 [22] 和 DocOwl-1.5 [10] 使用视觉语言方法和更大的模型规模，实现了更通用的截图到 Markdown 转换。OlmOCR [33] 基于 Qwen2-VL [42]，微调；该模型将 PDF 文件中的文本及其对应坐标作为锚点，纳入语言模型的提示中。LlamaParse[2] 和 Mistral OCR [3] 等闭源服务也提供 PDF 到 Markdown 转换。本文介绍了 EditTrans：它会编辑 Nougat、Kosmos-2.5 和 OlmOCR 的输入序列，以加快转换速度。

2.2 文档布局分析

得益于大规模文档布局数据集的支持，近期的文档布局分析（DLA）模型已变得愈加强大 [13,20,32,50] 计算机视觉模型能够从文档屏幕截图中提取布局 [19,44,47]。语言模型也被用于识别布局。LayoutLM [46] 及其变体 VILA [35] 是 Transformer 编码器模型，可根据文本及其二维坐标分析文档布局。LayoutLMv2 和 3 [12, 45] 还将视觉特征附加到 Transformer 编码器上。DiT [18] 通过无监督数据进行自回归训练。LILT [41] 将英语布局预训练扩展到了多种语言。ERNIE-Layout [31]引入了阅读顺序预测任务，提高了 DLA 模型的准确性。在本研究中，我们的可编辑文本识别模块基于 ERNIE-Layout 进行微调。

2.3 使用文本编辑模型进行文本生成

Transformer 解码器模型从头开始逐个词元生成输出，因此推理速度较慢。与解码器模型相比，文本编辑模型具有多项优势，包括更快的推理速度、更高的样本效率，以及对输出更好的控制能力和可解释性 [25]。LaserTagger [26] 通过三种操作实现句子融合任务：保留、删除、

[2] https://cloud.llamaindex.ai/parse [3] https://mistral.ai/news/mistral-ocr

以及 REPLACE。FELIX [24] 和 EdiT5 [23] 也能实现文本重排。PIE [1],、Seq2Edits [37], 和 GECToR [29]使用迭代式精炼方法编辑文本。有一篇博客文章[4] 介绍了 Google 搜索如何使用 EdiT5 [23] 纠正用户输入，并实现低延迟。此外，Locate&Edit [36] 等近期创新代表了一种用于受控文本生成的基于能量的方法：它可以与黑盒语言模型协同工作，首先识别与特定约束相关的文本片段，然后有选择地将其替换为更合适的表述，从而在满足所需约束的同时保留原文的核心语义。我们的工作将源自 PDF 的文本整理到编辑队列中，模拟文本编辑模型的行为。

2.4 中间填充

中间填充（FIM） [3] 是一种直接而有效的方法，可用于训练语言模型补全文档中间缺失的文本。大量评估表明，使用这种转换后的数据进行训练，不会损害模型原有的从左到右生成能力。FIM 技术已应用于多种自然语言处理任务，包括文档补全、代码编辑和写作辅助。目前，用于编程辅助的大语言模型服务采用了 FIM 范式，例如 Gemini Coder[5]，Claude Code[6]，和 DeepSeek-Coder [9]。我们任务中的编辑操作执行沿用了 FIM 范式。

3 方法

EditTrans 将学术文档转换流程精简为三个模块：

1. **可编辑文本识别：** EditTrans 首先对从 PDF 页面中提取的文本片段进行分类，并识别其中哪些部分可以复制。2. **编辑队列构建：** 随后，EditTrans 将分类后的文本片段组织成编辑操作队列，并为每项待执行的编辑操作确定停止准则。3. **编辑操作执行：** 对于每个需要编辑的文本片段，EditTrans 利用预训练的 PDF 到 Markdown 模型，采用 FIM 范式执行必要的编辑。

图 1 简要展示了如何从 PDF 复制文本，并将端到端 PDF 到 Markdown 模型的推理步骤保存下来，从而降低延迟。图 4 展示了 EditTrans 处理行内公式的示例。

[4] https://research.google/blog/grammar-checking-at-google-search-scale/ [5] https://geminicoder.org/ [6] https://docs.anthropic.com/en/docs/agents-and-tools/claude-code/overview

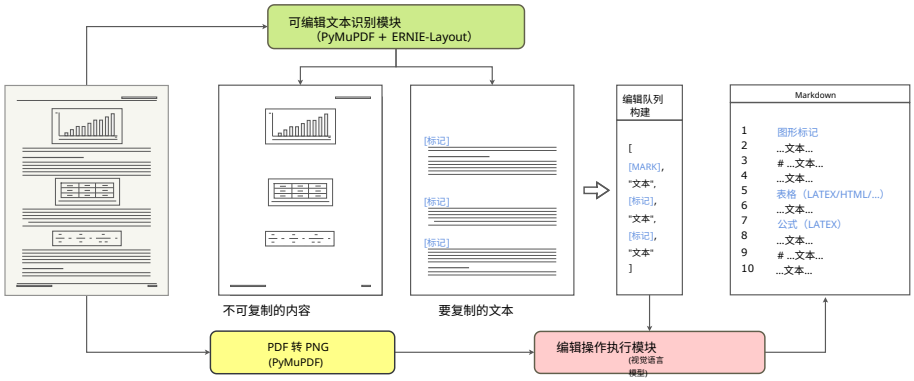


图 1. EditTrans 工作流。可编辑文本识别模块会检测文本片段是否可复制。编辑队列构建模块会构建编辑队列，其中 [标记] 用于启动生成。编辑操作执行模块会执行编辑：蓝色部分由采用中间填充范式的骨干视觉语言模型生成。

3.1 可编辑文本识别

受文档布局分析相关工作的启发，我们假设文本是否可复制与其布局信息高度相关。具体而言，我们提出如下观点：

1. 段落中密集的纯文本应完整保留。2. 数学公式、表格和标题等页面元素应进行修改，以符合 Markdown 格式规范。3. 页眉、页脚和页码等不传达相关内容的元素应从最终文档中排除。

遵循 VILA [35], ，我们假设文本在文本片段级别上的可编辑性是一致的。我们使用 PyMuPDF[7] 从 PDF 中提取文本片段级别的文本和边界框。随后，我们使用 PyMuPDF4LLM[8],的一种启发式方法来检测文本片段的阅读顺序。

我们使用 LoRA[11], 对 ERNIE 布局模型进行词元分类微调，并省略全局 1D 位置嵌入，以避免布局判断中可能产生的偏差 [40]。该模块将目标类别从 DLA 模型通常使用的广泛布局类别（例如，DocBank 最多使用 10 个类别 [20]）简化为 3 个类别。我们修改了 ERNIE-Layout 的分类头，使其预测标签为 KEEP、DELETE 或 INSERT_LEFT。

– 保留表示该文本片段应原样保留在 Markdown 输出中；– 删除表示该文本片段应被删除；

[7] https://github.com/pymupdf/PyMuPDF [8] https://pymupdf.readthedocs.io/en/latest/pymupdf4llm/

– INSERT_LEFT 表示应在此文本片段之前插入 PDF 到 Markdown 模型生成的触发标记。

这种方法与现有的文本编辑模型显著不同。本文中的编辑逻辑是删除不应复制的文本，然后插入解码器应生成的词元。例如，从 PDF 中删除数学公式，并在下一个可编辑文本片段之前生成一个等价表达式。

由于我们已针对词元分类对 ERNIE Layout 进行了微调，而且每个片段可能包含多个词元，因此采用投票分类器 [6] 来判定片段的预测结果。微调超参数的详细信息见附录 B。

3.2 编辑队列构建

获得片段级编辑标注后，我们可以将其转换为编辑队列Q ，以指示预训练的 PDF 到 Markdown 模型应编辑文本的哪些部分。每个编辑队列 Q都以编辑触发器开头，随后依次处理各个片段。我们遍历编辑标注中的每个片段。

如果下一个片段被预测为保留，我们就将片段文本添加到 Q。需要注意的是，如果该片段中的文本字符数非常少，我们就不添加它，而是让 PDF 到 Markdown 模型生成它，因为文本过短会难以匹配 PDF 到 Markdown 模型应停止生成的位置。然后，我们将该片段文本的首词与 PDF 到 Markdown 模型的输出进行匹配，以标记其停止生成并开始复制的位置。

如果下一个片段被预测为删除，则不向 Q添加任何内容。

如果下一个片段为插入_LFET，我们会先添加一个编辑触发器，然后将该片段的文本序列添加到 Q中。与 KEEP 片段类似，我们会将开头的几个词与 PDF 到 Markdown 模型中的编辑停止标记进行匹配。需要注意的是，如果该片段中文本字符的长度非常短（例如少于 5 个字符），我们不会添加它，而是期望 PDF 到 Markdown 模型生成该片段。这种启发式方法可以避免低效的碎片化生成，即模型频繁中断以复制很短的片段。

最后，我们会添加一个额外的编辑触发器，使 PDF 到 Markdown 模型能够生成句末令牌。

3.3 编辑操作执行

算法 1 简要展示了编辑操作的执行过程。在此步骤中，我们初始化一个空的词元序列 S ，并遍历编辑队列 Q。

如果 Q 中的下一个元素是编辑触发器，则将页面截图和S 输入 PDF 到 Markdown 模型。PDF 到 Markdown 模型的生成阶段包括中间填充 Markdown 任务，该任务会预测序列中的下一个令牌，直到满足停止条件。PDF 到 Markdown 模型的输出将添加到 S。

输入：编辑操作队列 Q、PDF 文档 D、Nougat 模型 $\mathcal{M}$**输出：**生成的 Markdown 序列 S初始化空令牌序列 S；从 D提取页面截图；**当** Q 不为空时**执行**$q \leftarrow$ 出队(Q)；**如果** q 是编辑触发器**则**将页面截图和 S 输入 $\mathcal{M}$；**当** $\mathcal{M}$ 尚未达到停止条件时**执行**生成下一个令牌并将其追加到 S；**结束 结束 否则如果** q 是待复制片段**则**对来自 q 的文本进行分词，并追加到 S；**结束 结束**将 S 反分词为 Markdown 格式；**返回** S；**算法 1：**编辑操作执行流程，该伪代码中的示例端到端 PDF 到 Markdown 模型是 Nougat。

如果队列中的下一个元素是待复制片段，我们会对该片段的文本进行词元化，并将其添加到 S 的末尾。

中间填充 *Markdown*。 在此模块中，预训练 PDF 模型以当前正在生成的令牌序列为输入，并开始自回归解码，直到触发停止条件。我们将停止条件设为下一个待复制片段的前 n 个词元或句末词元。

批次同步。 进行批次推理时，由于编辑队列 Q 中下一个待复制片段的长度不相等，执行每个编辑操作后，S 中序列的长度也不相等。我们对批次中的每个序列进行左侧填充，以确保批次内的操作同步。

跳过编辑操作。 在实践中，我们发现编辑操作队列中存在一些偏差。具体而言，可编辑文本识别（ETI）的识别结果可能不够可靠，因此可能会生成错误的编辑操作队列。为了尽量减少 ETI 错误的影响，我们在编辑操作执行模块中添加了跳过当前操作的功能。如果当前生成的最后 $n' > n$ 个词元与编辑操作队列中的下一个待复制片段一致，我们就从编辑操作队列 Q 中弹出下一个操作并继续执行。

n 和 n' 的取值（分别根据经验设为 3 和 5）代表一种权衡。取值较小可能导致生成过早终止，而取值较大则可能使模型忽略必要的小幅修改。

简而言之，我们从 PDF 中复制简单文本，并让 PDF 转 Markdown 模型负责生成复杂内容，例如公式和表格。最后，输出 S，并将其去词元化为 Markdown 格式。

EditTrans 的超参数见 C节。

4 实验

4.1 PDF 到 Markdown 模型

目前有三种开源的、带权重的 PDF 转 Markdown 模型。我们将它们用作骨干模型，以测试 EditTrans 的有效性。

1. Nougat [4], 我们使用 nougat-base[9]，它有 3.49 亿个参数。2. Kosmos-2.5[10] [22],，有 13.7 亿个参数。3. OlmOCR [33], 我们使用 olmOCR-7B-0225-preview[11]，有 82.9 亿个参数。

EditTrans 预计会生成与 Nougat、Kosmos-2.5 和 OlmOCR 相同的输出（即生成 Markdown）。EditTrans 需要以 PDF 页面作为输入，而 Nougat 和 Kosmos-2.5 需要 PDF 页面的截图；此外，OlmOCR 还需要 PDF 文件中的文本锚点来构建提示。

提示查找解码（PLD） [34]是一种通过识别并重用输入提示中出现的词元序列来加快大语言模型生成速度的技术。由于 OlmOCR 会将文件中的文本锚点用作提示的一部分，因此我们将其作为对比对象。Nougat 和 Kosmos-2.5 不适用于 PLD，因为它们的提示中没有文本。

4.2 数据集构建

由于目前尚无以完整 PDF 形式发布的现有数据集，我们从 arXiv 下载了 2023 年7月和8月论文的 LaTeX 源代码包。随后，我们使用一个框架 [7]将 LaTeX 编译为 PDF，并为页面上的每个元素标注语义标签、阅读顺序，以及对应于数学公式和表格的 LaTeX 代码。部分下载的论文源代码未能成功标注，因为其编写方式导致该框架无法解析。总计标注了 14,320 篇论文，共计 180,146 页。

从这些页面中提取文本片段，并根据上一步语义标注的结果，将其标记为 KEEP、DELETE 或 INSERT_LEFT。由于图和表格的标题顺序会被调整，因此我们将其标记为 DELETE。

数据集中排除了空白页或难以阅读的页面，例如包含整页图像、长表格或参考文献的页面。最后，我们构建了一个包含 180.146 页的数据集，每页均标注了

[9] https://huggingface.co/facebook/nougat-base [10] https://huggingface.co/microsoft/kosmos-2 5 [11]。 https://huggingface.co/allenai/olmOCR-7B-0225-preview

可复制的跨度级文本标签及其对应的边界框。我们的数据集布局丰富多样，涵盖单栏和多栏格式、复杂表格、大量数学方程以及嵌入式图，能够在简单布局之外进行稳健评估。我们随机划分数据集，训练集包含 162,127 页，测试集包含 18,019 页。该数据集中的绝大多数页面均为英文。

然后，我们为每页附上一个 Markdown 目标文本，以模仿 PDF 到 Markdown 模型将数学公式和表格作为 LaTeX 代码插入的风格。LaTeX 相当灵活，因为它允许用户自定义宏。因此，我们使用 LaTeXML[12] 对公式和表格的 LaTeX 代码进行规范化。

本文的方法从 PDF 文件中提取文本片段，因此需要访问学术论文的全文。由于 arXiv 不允许转载全文[13], ，我们发布了用于创建数据集的脚本以及数据集中论文的 arXiv 编号，以确保结果可复现。

4.3 特定模型的 PDF 到 Markdown 测试集构建

以往的工作中，由于 Nougat 和 Kosmos-2.5 均未发布其数据集，因此转换质量是在其私有数据集上评估的。另一方面，OlmOCR 采用人工评估。这些评估都难以复现。

我们从测试集中随机选取了 128 页，用于测试 PDF 到 Markdown 转换的速度和质量。此外，我们还分别从归类为经济学和量子物理的论文中随机选取了 128 页，以测试 EditTrans 在不同领域的表现。

实际上，我们发现，三个模型的输出格式并不相同，这取决于它们的训练策略。

1. Nougat 将 PDF 页面中的所有数学公式转换为 LaTeX 代码。随后，Nougat 将表格主体以及图和表的标题置于该页面 Markdown 的末尾。2. Kosmos-2.5 将 PDF 页面中的表格转换为 HTML 格式[14]，，并将表格及其标题置于页面开头。随后，Kosmos-2.5 将图的标题置于页面末尾。3. OlmOCR 会按照表格和图在页面中出现的顺序排列它们，并使用 Markdown 表格语法转换表格。

然而，这些差异并不会导致转换结果的可读性变差。鉴于 PDF 到 Markdown 转换任务具有相对较大的灵活性，我们针对每个模型的行为进行了适配，并为每个模型整理了相应的标准转换结果。

这为转换质量提供了公平且可复现的评判标准。为了也能公平地评估未来的 PDF 转 Markdown 模型，制作测试数据集的脚本已在 GitHub 上开源。

[12] https://math.nist.gov/~BMiller/LaTeXML/ [13] https://info.arxiv.org/help/license/reuse.html#full_文本 [14] 这一观察结果与其论文中所描述的不同 [22]。

4.4 评估指标

我们对每个模块进行了计时，以评估 EditTrans 所降低的转换延迟。遵循 Nougat 的做法，我们使用编辑距离 [17], BLEU [30],METEOR [2], 精确率、召回率和 F-1 来评估转换质量。

编辑距离 用于计算将一段文本转换为另一段文本所需的最少字符编辑操作（插入、删除、替换）次数，并以总字符数的比率表示，以便进行标准化比较。

BLEU是机器翻译任务中广泛使用的评估指标。BLEU 通过统计预测文本与参考文本之间重叠的词序列（n元语法）来评估文本相似度。

METEOR 是一种侧重召回率的翻译评估指标。它通过考虑同义词匹配和释义内容，衡量预测文本与参考文本之间的语义对齐程度。

精确率、召回率和 F-1 为生成文本预测的准确性和完整性提供了平衡的衡量方式。

5 结果

5.1 EditableText识别模型准确率

我们使用采用微平均的词元级 F1 分数评估分类器的性能。获得的词元级 F1 分数为 **0.963**。该分类器展现出高效的推理性能，每页仅需约 0.03 秒。

5.2 PDF 到 Markdown 转换延迟与质量

表 1 显示，EditTrans 可节省推理步数，从而降低延迟。总体而言，Kosmos-2.5 从 EditTrans 中获益最大：在量子物理数据集上，生成步骤最多减少 43.8%；在 arXiv 数据集上，至少减少 36.8%。相比之下，Nougat 获益最小，在 arXiv 数据集上的步数减少幅度低至 29.2%。总体来看，延迟的降低幅度与所节省的推理步数比例高度一致。此外，我们观察到，PLD 能够降低 OlmOCR 的延迟，但效率不如 EditTrans。

表 2 说明，EditTrans 通常能够改善 PDF 到 Markdown 转换的质量，或仅造成极小影响。就编辑距离而言，EditTrans 通常能够降低各数据集上的错误；但在 arXiv 数据集上，Kosmos-2.5 的编辑距离略有增加，增幅为 0.003。基于翻译的指标（BLEU 和 METEOR）差异可以忽略不计，变化均低于 0.01，表明 EditTrans 的实质性影响极小。应用 EditTrans 后，F1 分数大多有所提升，但 Nougat 在 arXiv 和经济学数据集上的情况除外，变化为 0.005。总体而言，这些结果表明，EditTrans 能够始终保持或略微提升转换质量，且不会产生任何显著的负面影响。我们未在表中加入 PLD 的对比结果，因为它不会改变生成的文本。

表 1。三个模型在三个测试数据集上的生成步骤和推理延迟。w/o ET 表示模型直接读取 PDF 文件并生成 Markdown 代码。w/ ET 表示借助 EditTrans 生成的模型。Ol mOCR 还比较了 PLD 的效率。

		指标			
模型	测试数据集	生成步骤	保存步骤	延迟（秒）	延迟降低
Nougat w/o ET	arXiv	926.82	-	6.37	-
Nougat（使用 ET）		656.28	29.2%	4.47	29.8%
Kosmos-2.5（不使用 ET）		879.28	-	11.62	-
Kosmos-2.5（使用 ET）		555.88	36.8%	7.24	37.7%
OlmOCR（不含 ET）		1148.47	-	34.47	-
OlmOCR（含 ET）		766.69	33.2%	22.91	33.5%
Nougat（无 ET）	经济学	817.62	-	5.63	-
Nougat（有 ET）		562.17	31.2%	3.84	31.7%
Kosmos-2.5（无 ET）		786.66	-	10.38	-
Kosmos-2.5（有 ET）		446.27	43.3%	5.81	44.0%
OlmOCR（不含 ET）		870.08	-	26.42	-
OlmOCR（含 ET）		561.82	35.4%	17.07	35.3%
不含 ET 的 Nougat	量子物理学	967.80	-	6.59	-
含 ET 的 Nougat		607.63	37.2%	4.12	37.4%
不含 ET 的 Kosmos-2.5		911.75	-	11.99	-
Kosmos-2.5（带 ET）		511.97	43.8%	6.66	44.5%
OlmOCR（不带 ET）		1174.38	-	35.73	-
OlmOCR（带 ET）		738.45	37.1%	22.32	37.5%

表 2. 三个模型在三个测试数据集上的 PDF 到 Markdown 转换质量。w/o ET 表示模型直接读取并生成 PDF 文件的截图。w/ ET 表示借助 EditTrans 生成的模型。

		指标					
模型	测试数据集	编辑距离 ↓	BLEU ↑	METEOR ↑	精确率 ↑	召回率 ↑	F1 ↑
Nougat 不使用 ET	arXiv	0.343	0.547	0.616	0.822	0.675	**0.720**
Nougat 使用 ET		**0.335**	0.542	0.619	0.801	0.676	0.715
Kosmos-2.5 不使用 ET		**0.384**	0.506	0.590	0.724	0.634	0.656
Kosmos-2.5 使用 ET		0.387	0.516	0.600	0.737	0.644	**0.666**
OlmOCR（不含 ET）		0.311	0.541	0.676	0.671	0.679	0.667
OlmOCR（含 ET）		**0.294**	0.532	0.683	0.698	0.699	**0**
Nougat（不含 ET）	经济学	0.283	0.645	0.710	0.838	0.759	**0.778**
Nougat（启用 ET）		**0.278**	0.625	0.701	0.819	0.746	0.762
Kosmos-2.5（未启用 ET）		0.320	0.590	0.699	0.751	0.738	0.726
Kosmos-2.5（启用 ET）		**0.312**	0.604	0.706	0.768	0.742	**0**
OlmOCR（未启用 ET）		0.281	0.594	0.742	0.707	0.737	0.712
OlmOCR（启用 ET）		**0.277**	0.584	0.734	0.725	0.760	**0**
Nougat 不使用 ET	量子物理学	0.356	0.528	0.580	0.838	0.667	0.729
Nougat 使用 ET		**0.349**	0.550	0.601	0.863	0.688	**0**
Kosmos-2.5 不使用 ET		0.384	0.528	0.600	0.778	0.660	0.696
Kosmos-2.5（含 ET）		**0.374**	0.545	0.616	0.794	0.671	**0.713**
OlmOCR（不含 ET）		0.294	0.573	0.699	0.715	0.707	0.704
OlmOCR（含 ET）		**0.257**	0.571	0.712	0.735	0.712	**0**

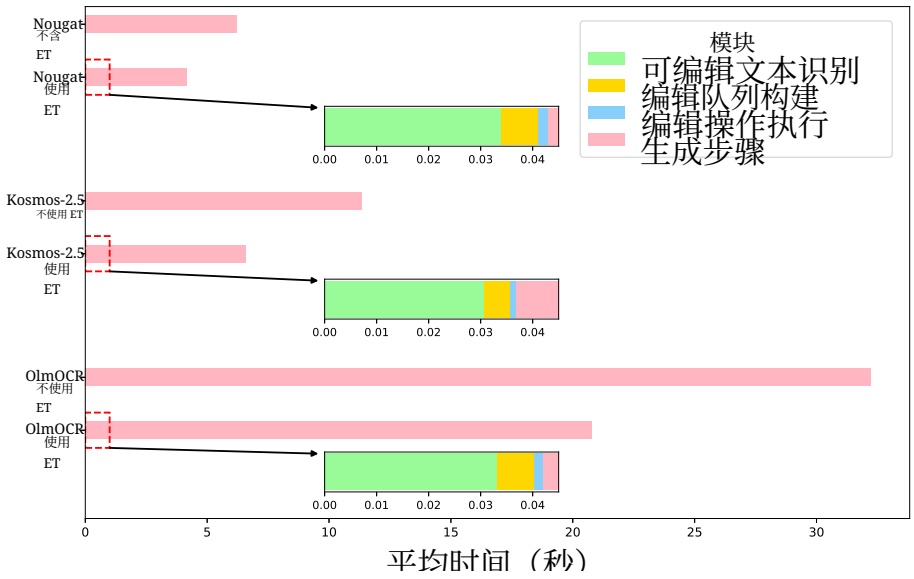


图 2. 各模块的执行时间开销。其中 ET 表示 EditTrans。带 ET 的 Generation Step 表示第 3.3 节所述的中间填充 Markdown 步骤，不带 ET 的 Generation Step 则表示模型的推理步骤。从图中可以看出，EditTrans 模块会增加一部分延迟（~0.04 秒），但与其在 Generation Step 中节省的时间相比，这部分延迟微不足道。

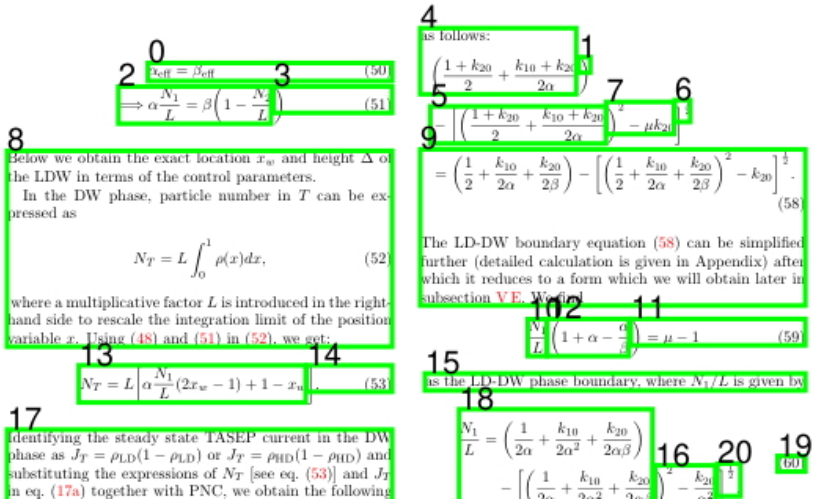


图 3. 构建编辑操作队列时遇到的边缘案例。绿色方框表示文本片段，数字表示 PyMuPDF4LLM 预测的阅读顺序。在此示例中，左侧的文本列本应从上到下读取，随后再读取右侧的文本列。但其阅读顺序被打乱，导致 EditTrans 在这种情况下无法发挥作用。

5.3 EditTran 各模块的延迟

我们通过消融研究分析了各个模块带来的执行时间开销。如图 2 所示，集成 EditTrans 会略微增加延迟。然而，与生成步骤节省的大量时间相比，这一开销微不足道。值得注意的是，这种额外延迟保持稳定，不会随着骨干模型规模增大而上升。

5.4 错误分析

*缓解 ETI 错误传播。*一个核心挑战是，上游 ETI 模块中的错误可能会向下游传播。我们专门设计了 *跳过编辑操作*机制（3.3）来应对这一问题。该功能起到自我纠正的作用。例如，如果 ETI 模块错误地将某个片段标记为需要生成，但骨干模型仍生成了后续 KEEP 片段的内容，我们的跳过机制就会检测到这种重叠，并阻止冗余的复制操作。这使处理流程能够从 ETI 错误中平稳恢复，从而增强其鲁棒性。

*垃圾进，垃圾出。*如图 3 所示，PyMuPDF4LLM 虽然总体可靠，但面对极其复杂的 PDF 布局时可能会遇到困难。这个示例属于极为罕见的边缘情况（占我们数据集页面的 $<0.03\%$），此时 PyMuPDF4LLM 用于片段提取的空间启发式规则会误判实际的阅读顺序。我们当前选择采用 PyMuPDF4LLM，是经过审慎权衡的决定：在绝大多数文档中，它的准确性通常较为稳健，处理效率也很高；而此类低频复杂案例则构成了挑战。因此，当这些罕见的阅读顺序错误确实发生时，EditTrans 生成的编辑操作队列会失效，导致这些特定页面的生成步骤无法减少。我们注意到，OlmOCR 的文本锚点提取方法似乎能够按照正确的阅读顺序提取文本。未来的研究中，我们将转而采用这种方法。

*词汇表越小越好。*在质量评估过程中，我们频繁发现 Kosmos-2.5 会错误表示行内数学公式的 LaTeX 代码。例如，根据 LaTeX 标准，符号 $\pm$ 应编码为 \pm，但 Kosmos-2.5 却输出 Unicode 字符 \u00b1。这个问题很可能是因为 Kosmos-2.5 使用了庞大的词汇表，而且与 Nougat 等模型相比，其预训练数据集可能未能有效过滤掉此类特定的编码错误。

*1024 个词元还不够。*Nougat 和 Kosmos-2.5 的最大生成长度设为 1024，而 OlmOCR 设为 2048。这是由于各模型的最大长度不同，且受到 GPU 内存限制。表 3 表明，1024 个词元的最大序列长度通常不足以完整转换一页科学文档。多达

80.5% 的页面因这一限制而遭到截断。尽管 EditTrans 有时可以通过复制跨度操作让单个输出略微超过这一限制，但这并不能从根本上解决截断问题。即使 OlmOCR 将词元上限提高到 2048，在 arXiv 数据集中的 10.2% 案例里仍然不够。因此，序列长度限制仍然是 PDF 到 Markdown 转换面临的重大挑战。此外，我们观察到，表格以及其次的复杂公式会消耗大量词元资源。专门针对表格制定一种语言模型友好型标准，或许可以缓解这一问题。

表 3. 预训练 PDF 到 Markdown 模型有时会生成达到最大序列长度限制的输出。这种情况下，页面末尾的文本会被截断，可能导致数据丢失和转换不完整。

	测试数据集		
模型	arXiv	经济学	量子物理
Nougat	67.2%	43.8%	80.5%
Kosmos-2.5	54.7%	37.5%	62.5%
OlmOCR	10.2%	1.6%	3.9%

专用模型更加稳定。 在评估 OlmOCR 的输出质量时，我们发现了若干一致性问题。(1) OlmOCR 偶尔会生成多余或含糊不清的图像说明；例如，![图 5：...](图像)符合 Markdown 格式，但包含语义上毫无意义的引用。(2) OlmOCR 在行内数学表达式中不一致地混用 $...$ 和 \(...\) 两种记法。虽然这两种格式在 LaTeX 中都有效，但混用会增加质量控制的难度。这些不一致主要源于 OlmOCR 基于 Qwen2-VL 进行微调；该模型是在高度多样化的常识数据集上预训练的，因此会带来不必要的差异性，并可能影响其在专门任务上的表现。

6 未来工作

我们发现，由于幻觉和重复文本等问题，现有的 PDF 到 Markdown 模型无法完整转换某些文档。即使使用 EditTrans，这些问题依然存在，因为它不会直接影响骨干模型的生成过程。LOCR [39] 通过利用视觉位置引导增强 Nougat 的输出，有效解决了这些问题，显著减少了幻觉和重复内容的出现。鉴于 LOCR 与 Nougat 具有互补性，它应能与 EditTrans 顺利集成。我们正密切关注 LOCR 的进展，并计划在其正式发布后将其与 EditTrans 结合。

另一项重大挑战是，目前的 PDF 到 Markdown 模型通常会在转换过程中排除图，而 ERNIE Layout 已证明

具备准确提取图形的能力。我们计划探索将图有效嵌入 Markdown 输出的方法。一种很有前景的方案借鉴了 Mistral OCR，它会将图像编码为 Base64 字符串，直接嵌入 Markdown 文档中。

此外，鉴于中间填充方法在代码生成方面取得了成功，大语言扩散模型 [28] 为未来研究开辟了一个引人入胜的方向。这种方法可能尤其适合 PDF 到 Markdown 任务，因为这类任务主要利用视觉信息和文本信息，而非仅依赖语言序列。

以用户为中心的评估若聚焦于转换后文档的可读性和学术实用性，将为了解我们方法的实际影响提供宝贵见解。基于树的编辑距离 [49] 也是一种可扩展至 Markdown 的评估指标。

7 结论

本文旨在解决端到端模型在将学术 PDF 转换为结构化 Markdown 这一任务中的严重低效问题。我们提出了 EditTrans，这是一种混合编辑生成框架，通过将简单的文本复制与复杂的生成任务分离，重新定义了这一问题。EditTrans 利用轻量级的布局感知分类器构建编辑队列，再引导功能强大的现成模型执行针对性的中间填充生成，从而显著提升了效率。全面的实验表明，在多个最先进的骨干模型上，延迟最多降低了 44.5%，解码步数减少了 43% 以上，同时最终输出的质量和保真度得以保持，甚至略有提升。

这项工作的意义远不止于加速处理。EditTrans 为数字图书馆和档案界提供了一条切实可行的途径，使文档处理能够扩展至大规模应用。它在不牺牲质量的前提下，大幅降低了转换的计算成本，使大规模学术收藏的高保真数字化——现代研究基础设施的基石——在经济和时间上都切实可行。这直接助力实现 FAIR 原则：将静态且难以解析的 PDF 文件转化为易于访问、可搜索且机器可操作的内容，从而提升其对研究人员、文献保护工作者及更广泛学术界的价值。

尽管我们承认当前实现存在一些局限，例如依赖上游阅读顺序工具，且重点关注文本内容，但 EditTrans 框架奠定了稳健且模块化的基础。其设计本身便支持未来的改进，包括集成更先进的版面分析和图形处理模块。总而言之，EditTrans 是迈向更智能、高效且可扩展的工作流程的务实而重要的一步，有助于在数字时代保存和传播学术知识。

致谢

本研究在研究项目 InsightsNet（insightsnet.org）框架内开展，该项目由德国联邦教育与研究部（BMBF）资助，资助编号为 01UG2130A。我们衷心感谢 Wolf-gang Stille 博士以及 hessian.AI 服务中心（由德国联邦教育与研究部 BMBF 资助，资助编号为 01IS22091）和 hessian.AI 创新实验室（由黑森州数字战略与创新部资助，资助编号为 S-DIW04/0013/003）提供的支持。我们感谢 Sabine Bartsch 博士对本研究的指导。我们也感谢匿名审稿人提出的真知灼见和建议，这些意见大大改进了本文。

参考文献

1. Awasthi, A., Sarawagi, S., Goyal, R., Ghosh, S., Piratla, V.：用于局部序列转换的并行迭代编辑模型。载于：Inui, K., Jiang, J., Ng, V., Wan, X.（编）《2019年自然语言处理经验方法会议与第九届国际自然语言处理联合会议（EMNLP-IJCNLP）论文集》。第 4260–4270 页。计算语言学协会，香港，中国（ 2019 年 11 月）。 https://doi.org/10.18653/v1/D19-1435，https://aclanthology.org/D19-1435/ 2. Banerjee, S., Lavie, A.：METEOR：一种用于机器翻译评估的自动指标，与人工判断的相关性更高。载于：Goldstein, J., Lavie, A., Lin, C.Y., Voss, C.（编）《机器翻译和/或摘要的内在与外在评估指标 ACL 研讨会论文集》。第 65–72 页。计算语言学协会，美国密歇根州安娜堡（2005年6月），https: //aclanthology.org/W05-0909/ 3. Bavarian, M., Jun, H., Tezak, N., Schulman, J., McLeavey, C., Tworek, J., Chen, M.：高效训练语言模型以实现中间填充（2022）4. Blecher, L., Cucurull, G., Scialom, T., Stojnic, R.：Nougat：面向学术文档的神经光学理解（2023）5. Cui, L., Xu, Y., Lv, T., Wei, F.：文档人工智能：基准、模型与应用（2021）6. Diem, M., Kleber, F., Sablatnig, R.：纸本文档片段的文本分类与文档版面分析。载于：2011年国际文档分析与识别会议（ICDAR）。第 854–858 页。IEEE，美国新泽西州皮斯卡塔韦（2011）。https://doi.org/10175 711092011./ICDAR... Duan, C., Tan, Z., Bartsch, S.：LaTeX rainbow：通用 LaTeX 到 PDF 文档语义与布局标注框架。载于：Ghosal, T., Grezes, F., Allen, T., Lockhart, K., Accomazzi, A., Blanco-Cuaresma, S.（编）《第二届科学出版物信息抽取研讨会论文集》。第 56–67 页。计算语言学协会，印度尼西亚巴厘岛（2023年11月）。https: //doi.org/10.18653/v1/2023.wiesp-1.8，https://aclanthology.org/2023.wiesp-1.8/ 8. Frankston, C., Godfrey, J., Brinn, S., Hofer, A., Nazzaro, M.：arXiv 上的 HTML 论文——其重要性及实现过程（2024）9. Guo, D., Zhu, Q., Yang, D., Xie, Z., Dong, K., Zhang, W., Chen, G., Bi, X., Wu, Y., Li, Y.K., Luo, F., Xiong, Y., Liang, W.：DeepSeek-Coder：大语言模型邂逅编程——代码智能的崛起（2024）

10. Hu, A., Xu, H., Ye, J., Yan, M., Zhang, L., Zhang, B., Li, C., Zhang, J., Jin, Q., Huang, F. 等：mPLUG-DocOwl 1.5：面向无 OCR 文档理解的统一结构学习。arXiv 预印本 arXiv:2403.12895（2024）11. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.：LoRA：大语言模型的低秩适配。载于：国际学习表征会议（ICLR）（2022），https://openreview.net/forum?id=nZeVKeeFYf912. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.：LayoutLMv3：通过统一文本与图像掩码对文档人工智能进行预训练。载于：第 30 届 ACM 多媒体国际会议（MM '22）论文集。第 4083–4091 页。计算机协会，美国纽约州纽约（2022）。https://doi.org/10.1145/3503161.354811213. Jaume, G., Kemal Ekenel, H., Thiran, J.P.：FUNSD：含噪扫描文档中表单理解的数据集。载于：2019年国际文档分析与识别研讨会（ICDARW）。第 2 卷，第 1–6 页。IEEE，美国新泽西州皮斯卡塔韦（2019）。https://doi.org/10.1109/ICDARW.2019.1002914. Kermitt2：GROBID（2008–2024），https://github.com/kermitt2/grobid15. Kim, G., Hong, T., Yim, M., Nam, J., Park, J., Yim, J., Hwang, W., Yun, S., Han, D., Park, S.：无 OCR 文档理解变换器。载于：《计算机视觉——ECCV 2022：第 17 届欧洲计算机视觉会议》。计算机科学讲义，第 13661 卷，第 466–483 页。Springer（2022）16. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J.E., Zhang, H., Stoica, I.：使用 PagedAttention 高效管理大语言模型服务的内存。载于：ACM SIGOPS 第 29 届操作系统原理研讨会（SOSP '23）论文集。计算机协会，美国纽约州纽约（2023）17. Levenshtein, V.I.：能够纠正删除、插入和反转的二进制码。《苏联物理学文献汇编》10(8), 707–710（1966年2月）18. Li, J., Xu, Y., Lv, T., Cui, L., Zhang, C., Wei, F.：DiT：文档图像变换器的自监督预训练。载于：第 30 届 ACM 多媒体国际会议（MM '22）论文集。第 3530–3539 页。计算机协会，美国纽约州纽约（2022）。https://doi.org/10.1145/3503161.354791119. Li, K., Wigington, C., Tensmeyer, C., Zhao, H., Barmpalios, N., Morariu, V.I., Manjunatha, V., Sun, T., Fu, Y.：跨域文档目标检测：基准套件与方法。载于：2020年 IEEE/CVF 计算机视觉与模式识别会议（CVPR）论文集。第 12912–12921 页。IEEE，美国新泽西州皮斯卡塔韦（2020）。https://doi.org/10.1109/CVPR42600.2020.0129320. Li, M., Xu, Y., Cui, L., Huang, S., Wei, F., Li, Z., Zhou, M.：DocBank：文档版面分析的基准数据集。载于：Scott, D., Bel, N., Zong, C.（编）《第 28 届计算语言学国际会议论文集》。第 949–960 页。国际计算语言学委员会，西班牙巴塞罗那（线上）（2020年12月）。https://doi.org/10.18653/v1/2020.coling- main.82，https://aclanthology.org/2020.coling-main.82/21. Loshchilov, I., Hutter, F.：解耦权重衰减正则化。载于：国际学习表征会议（ICLR）（2019），https://openreview.net/forum?id=Bkg6RiCqY722. Lv, T., Huang, Y., Chen, J., Cui, L., Ma, S., Chang, Y., Huang, S., Wang, W., Dong, L., Luo, W. 等：Kosmos-2.5：多模态读写模型。arXiv 预印本 arXiv:2309.11419（2023）

23. Mallinson, J., Adamek, J., Malmi, E., Severyn, A.：EdiT5：采用 T5 热启动的半自回归文本编辑。载于：Goldberg, Y., Kozareva, Z., Zhang, Y.（编）计算语言学协会年会论文集：EMNLP 2022。第 2126–2138 页。计算语言学协会，阿布扎比，阿拉伯联合酋长国（2022 年 12 月）。https://doi.org/10.18653/v1/2022.findings-emnlp.156, https://aclanthology.org/2022.findings-emnlp.156/24. Mallinson, J., Severyn, A., Malmi, E., Garrido, G.：FELIX：通过标注和插入实现灵活文本编辑。载于：Cohn, T., He, Y., Liu, Y.（编）计算语言学协会年会论文集：EMNLP 2020。第 1244–1255 页。计算语言学协会，线上（2020 年 11 月）。https://doi.org/10.18653/v1/2020.findings-emnlp.111，https://aclanthology.org/2020.findings-emnlp.111/25. Malmi, E., Dong, Y., Mallinson, J., Chuklin, A., Adamek, J., Mirylenka, D., Stahlberg, F., Krause, S., Kumar, S., Severyn, A.：使用文本编辑模型生成文本。载于：Ballesteros, M., Tsvetkov, Y., Alm, C.O.（编）2022年北美计算语言学协会年会论文集：人类语言技术：教程摘要。第 1–7 页。计算语言学协会，西雅图，美国（2022 年 7 月）。https://doi.org/10.18653/v1/2022.naacl-tutorials.1，https://aclanthology.org/2022.naacl-tutorials.1/26. Malmi, E., Krause, S., Rothe, S., Mirylenka, D., Severyn, A.：编码、标注、实现：高精度文本编辑。载于：Inui, K., Jiang, J., Ng, V., Wan, X.（编）2019年自然语言处理实证方法会议与第九届国际自然语言处理联合会议（EMNLP-IJCNLP）论文集。第 5054–5065 页。计算语言学协会，香港，中国（2019 年 11 月）。https://doi.org/10.18653/v1/D19-1510, https://aclanthology.org/D19-1510/27. Mindee：docTR：文档文本识别（2021 年），https://github.com/mindee/doctr28. Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.R., Li, C.：大语言扩散模型（2025 年） 29. Omelianchuk, K., Atrasevych, V., Chernodub, A., Skurzhanskyi, O.：GECToR——语法纠错：标注，而非改写。载于：Burstein, J., Kochmar, E., Leacock, C., Madnani, N., Pilán, I., Yannakoudakis, H., Zesch, T.（编）第十五届自然语言处理在教育应用中的创新研讨会论文集。第 163–170 页。计算语言学协会，西雅图，美国华盛顿州 → 线上（2020 年 7 月）。https://doi.org/10.18653/v1/2020.bea-1.16, https://aclanthology.org/2020.bea-1.16/30. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.：BLEU：一种机器翻译自动评估方法。载于：Isabelle, P., Charniak, E., Lin, D.（编）计算语言学协会第四十届年会论文集。第 311–318 页。计算语言学协会，费城，宾夕法尼亚州，美国（2002 年 7 月）。https://doi.org/10.3115/1073083.1073135, https: //aclanthology.org/P02-1040/31. Peng, Q., Pan, Y., Wang, W., Luo, B., Zhang, Z., Huang, Z., Cao, Y., Yin, W., Chen, Y., Zhang, Y., Feng, S., Sun, Y., Tian, H., Wu, H., Wang, H.：ERNIE-layout：利用版面知识增强预训练，实现富视觉文档理解。载于：Goldberg, Y., Kozareva, Z., Zhang, Y.（编）计算语言学协会年会论文集：EMNLP 2022。第 3744–3756 页。计算语言学协会，阿布扎比，阿拉伯联合

酋长国（2022 年 12 月）。https://doi.org/10.18653/v1/2022.findings-emnlp.274, https:// aclanthology.org/2022.findings-emnlp.274/ 32. Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A.S., Staar, P.：DocLayNet：一个由人工标注的大型文档布局分割数据集。载于：第 28 届 ACM SIGKDD知识发现与数据挖掘大会（KDD '22）论文集。第 3743–3751 页。计算机协会，美国纽约州纽约（20223539043 331011453534678）。https://doi.org/./.. Poznanski, J., Borchardt, J., Dunkelberger, J., Huff, R., Lin, D., Rangapur, A., Wilhelm, C., Lo, K., Soldaini, L.：olmOCR：利用视觉语言模型解锁 PDF 文件中的数万亿词元（2025 年） 34. Saxena, A.：提示查找解码（2023 年 11 月）, https://github.com/apoorvumang/prompt-lookup-decoding/ 35. Shen, Z., Lo, K., Wang, L.L., Kuehl, B., Weld, D.S., Downey, D.：VILA：利用视觉布局组改进科学类 PDF文档的结构化内容提取。《计算语言学协会汇刊》**10**，376–392（2022 年）。https://doi.org/10.1162/tacl_a_00466，https://aclanthology.org/2022.tacl-1.22/ 36. Son, H.R., Lee, J.Y.：Locate&Edit：基于能量的文本编辑，实现高效、灵活且忠实的可控文本生成（2024 年） 37. Stahlberg, F., Kumar, S.：Seq2Edits：使用跨度级编辑操作进行序列转导。载于：Webber, B., Cohn, T., He, Y., Liu, Y.（编）2020年自然语言处理实证方法会议（EMNLP）论文集。第 5147–5159 页。计算语言学协会，线上（2020 年 11 月）。https:// doi.org/10.18653/v1/2020.emnlp-main.418, https://aclanthology.org/2020.emnlp- main.418/ 38. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.：RoFormer：采用旋转位置嵌入的增强型 Transformer。《神经计算》**568**（2024 年 2 月）。https://doi.org/10127063 3910162023./j.neucom... Sun, Y., Zhou, D., Lin, C., He, C., Ouyang, W., Zhong, H.S.：LOCR：用于光学字符识别的位置引导型 Transformer（2024 年） 40. Tu, Y., Guo, Y., Chen, H., Tang, J.：LayoutMask：在文档理解的多模态预训练中增强文本与布局交互。载于：Rogers, A., Boyd-Graber, J., Okazaki, N.（编）计算语言学协会第六十一届年会（第一卷：长篇论文）论文集。第 15200–15212 页。计算语言学协会，多伦多，加拿大（2023 年 7 月）。https: //doi.org/10.18653/v1/2023.acl-long.847, https://aclanthology.org/2023.acl-long.847/ 41. Wang, J., Jin, L., Ding, K.：LiLT：一种简单而有效的语言无关布局 Transformer，用于结构化文档理解。载于：Muresan, S., Nakov, P., Villavicencio, A.（编）计算语言学协会第六十届年会（第一卷：长篇论文）论文集。第 7747–7757 页。计算语言学协会，都柏林，爱尔兰（2022 年 5 月）。https://doi. org/10.18653/v1/2022.acl-long.534, https://aclanthology.org/2022.acl-long.534/ 42. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., Lin, J.：Qwen2-VL：增强视觉语言模型在任意分辨率下对世界的感知（2024 年） 43. Wilkinson, M.D., Dumontier, M., Aalbersberg, I.J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.W., da Silva Santos, L.B., Bourne, P.E., Bouwman, J., Brookes, A.J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C.T., Finkers, R., Gonzalez-Beltran, A., Gray, A.J., Groth, P., Goble, C.,

Grethe, J.S., Heringa, J., 't Hoen, P.A., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S.J., Martone, M.E., Mons, A., Packer, A.L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.A., Schultes, E., Sengstag, T., Slater, T., Stra wn, G., Swertz, M.A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J., Mons, B.：科学数据管理与管理规范的FAIR指导原则。Scientific Data**3**(1) (2016年3月)。https://doi.org/10.1038/sdata.2016.18，http://dx.doi.org/10.1038/sdata.2016.1844. Wu, X., Hu, Z., Du, X., Yang, J., He, L.：通过动态残差特征融合进行文档版面分析。载于：2021年IEEE国际多媒体与博览会会议（ICME）。第1–6页。IEEE，美国新泽西州皮斯卡塔韦（2021年）。https://doi.org/ 10.1109/ICME51207.2021.9428465 45. Xu, Y., Xu, Y., Lv, T., Cui, L., Wei, F., Wang, G., Lu, Y., Florencio, D., Zhang, C., Che, W., Zhang, M., Zhou, L.：LayoutLMv2：面向视觉丰富文档理解的多模态预训练。载于：Zong, C., Xia, F., Li, W., Navigli, R.（编），第59届计算语言学协会年会暨第11届国际自然语言处理联合会议论文集（第1卷：长篇论文）。第2579–2591页。计算语言学协会，线上（2021年8月）。https://doi.org/10.18653/v1/2021.acl-long.201，https://aclanthology.org/2021.acl-long.201/ 46. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.：LayoutLM：面向文档图像理解的文本与版面预训练。载于：第26届ACM SIGKDD国际知识发现与数据挖掘会议（KDD '20）论文集。第1192–1200页。计算机协会，美国纽约州纽约（2020年）。https://doi.org/10.1145/3394486.3403172 47. Yang, H., Hsu, W.H.：使用循环卷积神经网络对科学文献进行基于视觉的版面检测。载于：2020年第25届国际模式识别会议（ICPR）。第6455–6462页。IEEE，美国新泽西州皮斯卡塔韦（2021年）。https://doi.org/10.1109/ICPR48806.2021.9412557 48. Zheng, L., Yin, L., Xie, Z., Sun, C., Huang, J., Yu, C.H., Cao, S., Kozyrakis, C., Stoica, I., Gonzalez, J.E., Barrett, C., Sheng, Y.：SGLang：高效执行结构化语言模型程序。载于：第38届神经信息处理系统年会（2024），https://openreview.net/forum?id=VqkAKQibpq 49. Zhong, X., ShafieiBavani, E., Jimeno Yepes, A.：基于图像的表格识别：数据、模型与评估。载于：计算机视觉——ECCV 2020：第16届欧洲会议，英国格拉斯哥，2020年8月23–28日，论文集，第XXI部分。第564–580页。施普林格出版社，瑞士尚（2020年）。https://doi.org/10.1007/978-3-030-58589-1_34 50. Zhong, X., Tang, J., Jimeno Yepes, A.：PubLayNet：迄今最大的文档版面分析数据集。载于：2019年国际文档分析与识别会议（ICDAR）。第1015–1022页。IEEE，美国新泽西州皮斯卡塔韦（2019年）。https://doi.org/10.1109/ICDAR.2019.00166 51. Zhu, W., Sokhandan, N., Yang, G., Martin, S., Sathyanarayana, S.：DocBed：适用于复杂版面的多阶段OCR解决方案。AAAI人工智能会议论文集**36**(11)，12643–12649（2022年6月）。https://doi.org/10.1609/aaai.v36i11.21539，https://ojs.aaai.org/index.php/AAAI/article/view/21539

一个示例

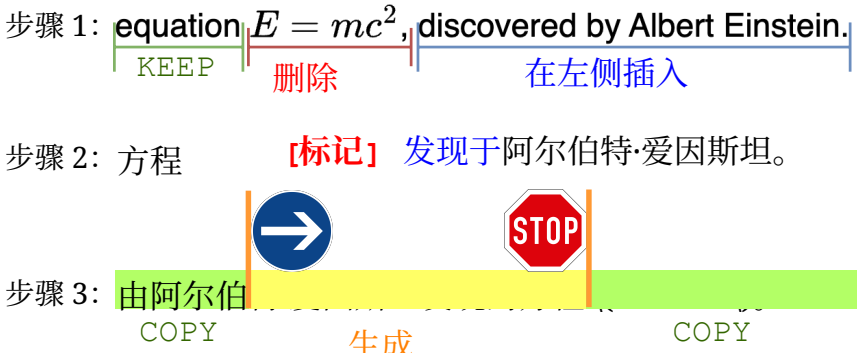


图 4. Edit Trans 处理行内公式的示例。步骤 1：检测该文本片段是否可复制。步骤 2：构建编辑队列，其中 [标记] 用于启动生成，而 蓝色单词 用作停止信号。步骤 3：执行编辑：绿色部分会被复制，而 黄色部分由骨干模型通过中间填充范式生成。

B 微调 ERNIE Layout 的详细信息

* 基础模型：`ERNIE-Layout-PyTorch`[15]* 旋转位置编码 [38] 实现：https://github.com/NormXU/ERNIE-Layout-Pytorch* 批次大小：48 * 训练轮数：10 * 权重衰减： 1×10^{-5}* 丢弃率：0.1 * 优化器：AdamW [21]

- 学习率： 2×10^{-5}
- ϵ： 1×10^{-6}

* LoRA：

- 秩：256
- α：256 * 参数总量：291,131,460 * 可训练参数：9,457,154 (3.25%)

* ERNIE-Layout 模型在一台 $1\times$A100 机器上微调了 12 小时。

[15] https://huggingface.co/Norm/ERNIE-Layout-Pytorch

C 实现细节

提示查找解码的查找词元数量设为 3。我们的实验采用了 Transformers 库的实现，因为它的实现最简单。我们计划改用 vLLM [16]和 SGLang [48]以提升速度，但这需要进行更多兼容性调整。

D 局限性

受 ERNIE 布局模型的限制，我们的方法会将编辑序列输入分块，长度最多为 1024 个词元，但我们发现许多页面的词元数量超过了这一上限。

整页公式和表格无法从我们的方法中受益。由于同步开销，EditTrans 在批量生成时可能效率较低，我们正在开发一种更灵活的方法。