题目

摘要

随着数字出版物的普及，对自动化文档布局生成器的需求日益增长。本文提出了一种基于 Transformer 的自动文档布局生成器，该生成器能够根据用户的具体需求生成符合阅读习惯和美学标准的文档布局。本研究的核心在于开发一系列创新算法，包括一个用于优化阅读顺序的分割递归算法，一个调整布局元素体积和长宽比的映射方法，以及一个新颖的布局采样策略。此外，我们还开发了一个直观的图形用户界面，显著提升了模型的易用性和互动性。实验结果表明，我们的模型在不同类型的文档布局任务上均表现出色，能够灵活适应各种布局要求。本文的研究为自动化文档布局设计提供了新的视角，并为未来的研究方向奠定了基础

关键词：自动文档布局，Transformer 模型，阅读顺序优化，布局采样策略，图形用户界面

1 引言

随着信息时代的到来，文档布局的设计变得越发重要，它不仅影响信息的呈现方式，还直接关系到信息的接收效果和用户体验。传统的文档布局设计依赖于设计师的经验和直觉，这种方法虽然能够创造出符合人类美学的布局，但却存在效率低下和难以处理大量数据的问题。随着人工智能和深度学习技术的发展，基于这些技术的布局生成器显现出巨大的潜力，特别是在处理大规模和复杂布局设计任务时。

近年来，深度学习在图像识别、语言处理等领域取得了显著成就，其在文档布局生成领域的应用也日益受到关注。相比于传统方法，基于深度学习的布局生成器能够从大量数据中学习布局规则和特征，自动化地生成高质量的布局方案。这种方法不仅提高了设计效率，还能够创造出传统方法难以实现的创新布局。

本文中的图片（见图 1）展示了一个复杂的文档布局示例，其中包含了多种元素如文本、标题、表格、图形和列表。文档布局的目的是为了创建一种视觉层次感，通过不同颜色的标记区分了不同类型的元素：文本（绿色框）、标题（红色框）、表格（蓝色框）、图形（黄色框）和列表（紫色框）。这种布局使得信息按照一定的逻辑顺序呈现，同时也允许读者通过不同的视觉线索来快速找到他们感兴趣的信息。在设计文档布局时，重要的是要确保所有元素都有足够的空间，并以一种有助于目标读者群理解和记忆的方式组织。

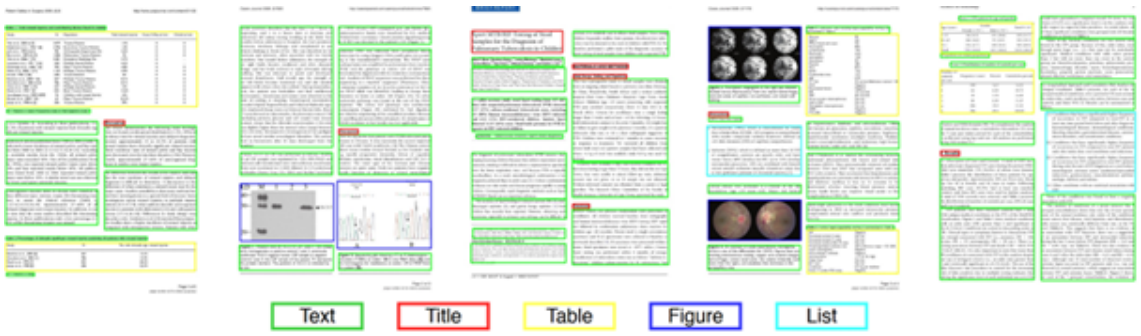


图 1. 复杂文档布局示例。不同颜色的框标识了文档中的不同元素类型，展示了布局设计中元素如何被组织和呈现。

在基于 Transformer 的文档布局生成器中，模型将学习如何从大量此类示例中识别和应用布局规则，以自动生成具有良好视觉层次感和读者友好性的文档布局。

本文提出的基于 Transformer 的文档布局生成器，旨在解决传统布局设计方法中的局限性，同时充分利用深度学习的优势。我们的方法不仅能够根据给定条件自动生成符合阅读习惯和美学要求的文档布局，而且能够对布局元素的大小和形状进行灵活控制，增加布局的多样性和可扩展性。此外，本文的方法还能够处理复杂的条件，如离群值和噪声，提高了模型的鲁棒性和泛化能力。

综上所述，本研究的意义在于提供了一种新颖的文档布局生成方法，不仅能够提高布局设计的效率和质量，还为文档生成领域带来了新的研究方向和技术突破。

2 相关工作

文档布局生成的研究可分为两大类：布局合成与基于 Transformer 的模型应用。

2.1 布局合成

布局合成旨在自动化地创造美观、易读的图形元素排列。早期工作依赖于规则或优化方法，例如布局模板和图文配对，但这些方法在处理复杂布局时受限。随着深度学习的发展，生成对抗网络（GANs）[3] 和变分自编码器（VAEs）[?] 开始用于布局生成，提高了自动生成高质量和逼真布局的能力。LayoutGAN [8] 和 LayoutVAE [5] 作为代表，引入了基于点的布局表示，注重元素的位置和类别，但忽视了细节如形状和大小。近期，神经设计网络（NDN）[7] 采用竞争性 VAEs 对元素关系和约束建模，使用边界框表示方法，但仍未解决重叠或遮挡元素的处理问题。

2.2 Transformer 和掩码语言模型

Transformer 模型 [9] 在自然语言处理领域极为关键，特别是在处理长序列文本方面。它基于注意力机制，能够并行处理上下文相关信息，优于循环神经网络（RNN）[?] 的序列处理。掩码语言模型如 BERT [2] 通过预训练来捕捉复杂的词依赖关系，进一步提升下游任务性能。尽管 Transformer 在语言任务上的效果显著，但其在生成任务上的应用仍在探索中。最新的

变分 Transformer 网络（VTN）[1] 和基于自注意力的布局生成模型 [4] 均尝试在无条件布局生成中学习全局设计规则，展示出高质量的布局生成潜力。

本文的贡献在于结合了 Transformer 的模型性能与掩码语言模型的预训练优势，提出了一种新的布局生成器，特别针对文档布局的生成进行了优化。通过引入分步采样策略，本文的方法不仅适应了文档布局的特性，而且实现了高质量且符合视觉感知规律的布局 [6]。

3 本文方法

与之前的研究相似，本研究将文档布局中的元素简化为轴对齐包围盒（Axis-Aligned Bounding Box, ABB），忽略倾斜和非矩形的形状。对于布局元素，我们关注五个主要信息：类别 c，位置 (x, y)，以及尺寸 (w, h)。所有文档尺寸被归一化到统一的矩形尺度。

3.1 布局参数表示

为了适应各种尺寸的文档，我们提出了一个包含 7 个属性的对象表示，$O = (c, x, y, w, h, a, r)$，其中 $c \in C$ 表示类别，$(x, y) \in \mathbb{R}^2$ 表示左上角位置，$w, h \in \mathbb{R}$ 表示宽度和高度，$a \in \mathbb{R}$ 和 $r \in \mathbb{R}$ 是新增的属性，分别代表体积和长宽比。这些浮点值通过 8 位均匀量化被离散化。例如，x 坐标量化后的范围是 $\{x \mid x \in \mathbb{Z}, 0 \leq x \leq 2^8\}$。布局 L 由 K 个元素表示为序列：

$$L = (O_{\pi_1}; O_{\pi_2}; \ldots; O_{\pi_K}, \text{eos}) \tag{1}$$

其中 $O_i = (c_i, x_i, y_i, w_i, h_i, a_i, r_i)$ 和 eos 是序列结束的特殊标记。π 表示元素的阅读顺序。

3.2 系统框架

本文模型借鉴了 BLT [6] 中的训练策略，该策略由 BERT [2] 启发，在训练过程中随机选择属性子集并用特殊的 <MASK> 标记替换，优化模型预测这些掩码属性。给定布局集 D，我们的目标是最小化如下损失函数：

$$\mathcal{L} = -\mathbb{E}_{L \in D, i \in M} \log p(L_i^* \mid L^M) \tag{2}$$

其中 L^M 是被掩码后的序列，M 是被掩码的位置集合，L_i^* 是模型的预测。

3.3 关键功能

在本文的方法中，我们整合了几个关键功能以提高文档布局生成的准确性和实用性。首先，我们设计了一种阅读顺序排序算法，通过搜索和递归切割布局来确定最优的阅读顺序。这个步骤对于模拟人类如何浏览文档至关重要。其次，我们引入了体积 a 和长宽比 r 的表示，这允许模型学习和控制布局元素的大小和形状，从而能够生成更丰富和多样的布局。此外，为了解决传统方法中常见的布局空白或重叠问题，我们提出了一种新颖的布局采样算法，该算法从两侧向中间分步替换标记，以优化整体布局的质量。最后，为了增加模型的可访问性和用户友好性，我们开发了一个图形界面，用户可以通过它轻松地进行条件生成和结果渲染，这大大提高了模型的实用性。

4 复现细节

4.1 与已有开源代码对比

本研究的实现是基于 layoutformer 论文提供的 PyTorch 代码框架。尽管 BLT 模型的原始实现是基于 TensorFlow，我们选择了 layoutformer 的 PyTorch 实现作为起点，进而实现了 BLT 模型。这一转变是基于 PyTorch 在某些方面如动态图执行、更直观的错误消息和广泛的社区支持方面的优势。在此过程中，我们不仅迁移了模型结构，还针对文档布局生成的特定需求进行了深入的定制和优化。我们的贡献在于成功地将 BLT 的核心概念融入到一个不同的代码基础中，同时保留并增强了原有模型的功能

4.2 实验环境搭建

模型的训练和测试都在个人电脑上完成，配置如下：AMD Ryzen 7 5800H 处理器、NVIDIA GeForce RTX 3070 Laptop GPU（8GB VRAM）、16GB DDR4 3200MHz RAM。程序开发使用的是 Visual Studio Code，利用 pytorch 机器学习框架进行模型训练，用户界面则是基于 pyside6（Qt for Python）框架构建。

模型的训练数据来源于 PubLayNet，一个包含超过 33 万个机器注释科学文档的数据集，涵盖文本、标题、图片、列表和表格等五个类别。为了提高训练效率，我们剔除了包含出界元素的布局样本，并且只选取了元素数量较少的 99% 的数据进行训练。模型结构配置为 4 层 Transformer，每层 8 个注意力头，嵌入维度为 512，隐藏层维度为 2,048。训练使用了 Adam 优化器，参数设置为 $\beta_1 = 0.9$ 和 $\beta_2 = 0.98$，批次大小为 64。

4.3 界面分析与使用说明

我们开发了一个图形用户界面（GUI）（见图 2），以便用户可以直观地与布局生成器交互，如附图所示。该界面提供了一个交互式的布局编辑器，用户可以在其中定义布局元素的属性，如类别、位置和尺寸。此外，用户还可以通过界面直接观察到布局生成器的输出，实时调整参数以获得最佳布局效果。这提高了模型的可访问性，并降低了对编程知识的要求。

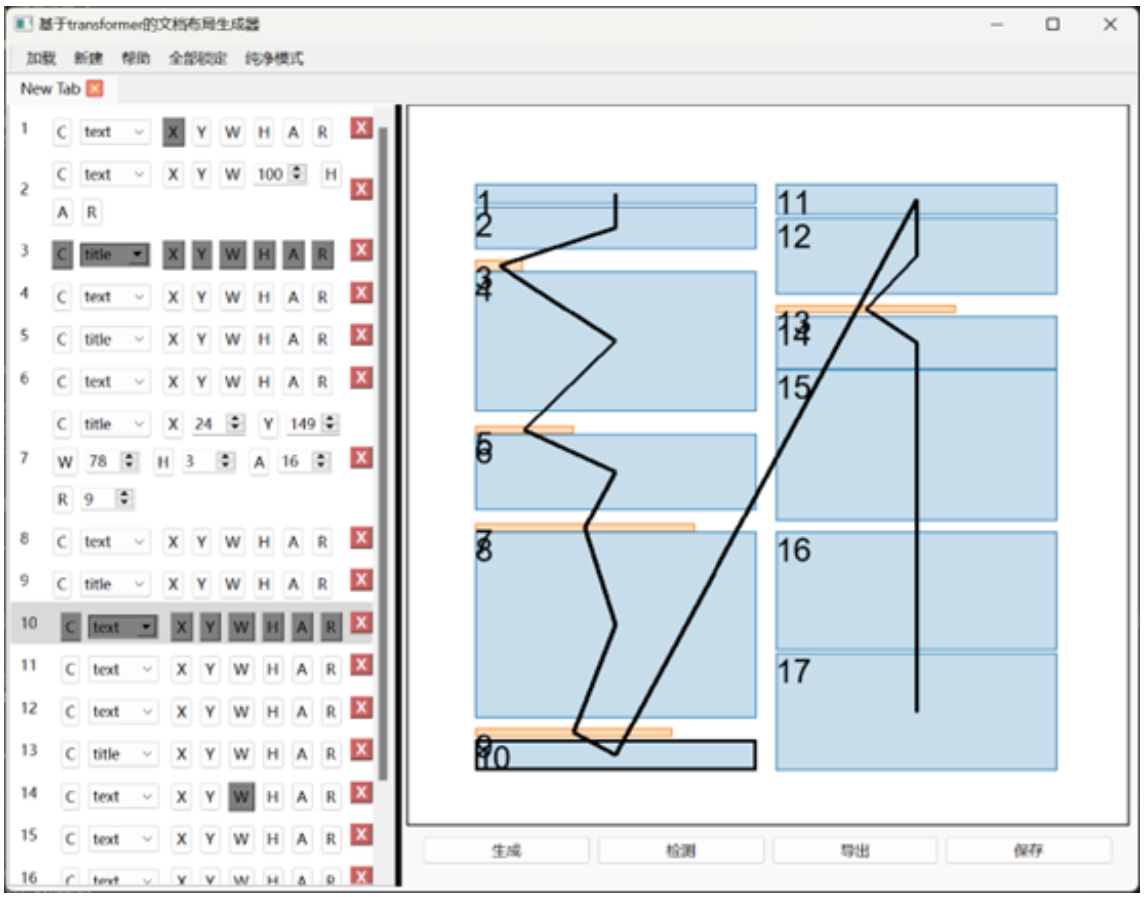


图 2. 开发的操作界面，展示了布局生成过程中的用户交互和实时布局调整。

4.4 创新点

在本研究中，我们实施了几项关键的创新点，旨在增强模型的性能和用户体验。首先，我们引入了一个精心设计的算法来精确地确定文档布局的阅读顺序。该算法不仅能搜索出最优布局切割方案，还能递归地处理切割后的子布局，以确保最终布局的阅读流畅性和逻辑性。

接着，我们对模型进行了拓展，使其能够掌握和控制布局元素的体积与长宽比，引入新的属性 a 和 r。这一点是通过对现有模型的进一步训练和调整实现的，以学习元素的尺寸和形状之间的微妙关系，从而丰富最终布局的多样性和逼真性。

此外，我们还开发了一种创新的布局采样策略，这种策略巧妙地从布局的两端开始，分步骤向中心替换标记。这一方法能有效避免生成过程中可能出现的空白区域或元素重叠，确保了生成布局的整体质量。

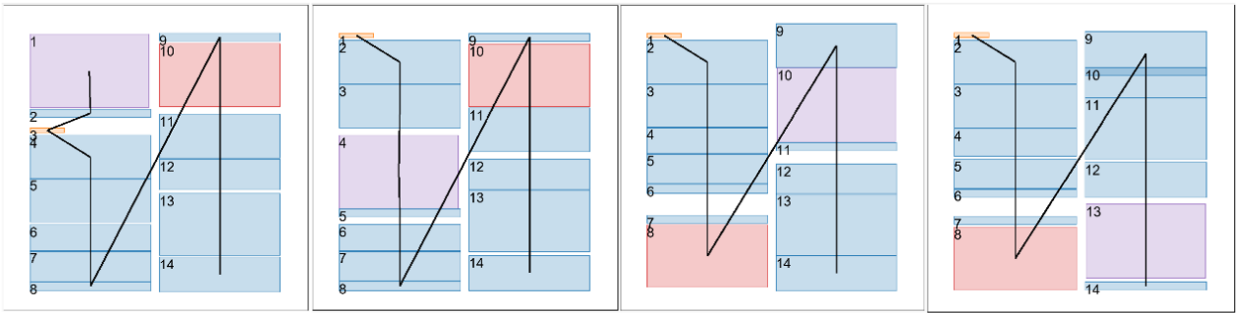
最后，为了提高模型的互动性和易用性，我们构建了一个直观的用户界面。该界面不仅让用户能够轻松地定义和修改布局元素的属性，还能实时地观察布局生成的过程和结果，使用户能够无需深入了解底层算法即可生成高质量的文档布局。

5 实验结果分析

本部分详细分析了模型在不同任务上的表现和结果。通过一系列的实验，我们评估了模型在顺序修改、体积调整、长宽比改变以及布局调整等方面的性能。

5.1 顺序修改

如图 3所示，我们展示了模型对元素顺序修改的能力。图 3(a) 展示了初始布局，其中首个元素是图片（紫色方框），紧随其后的是图片标注（蓝色方框）。在图 3(b)-(d) 中，我们在保持所有元素体积不变的前提下，逐渐调整图片和标注的顺序。模型成功地重新排列了元素，以适应这些变化。

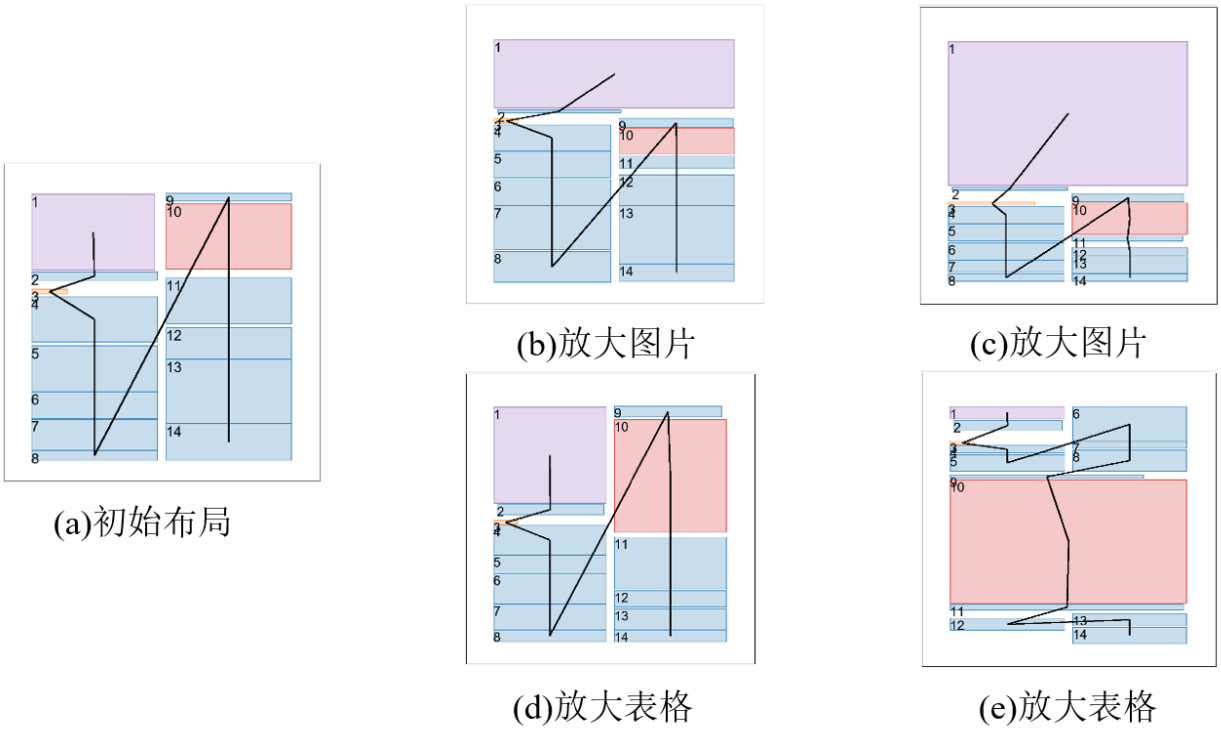


(a)初始布局 (b)顺序 1 (c) 顺序 2 (d) 顺序 3

图 3. 顺序修改示例。

5.2 体积修改

图 4展示了模型在体积修改任务上的表现。如图 4(a) 所展示的原始布局中，包含一个图片（紫色方框）和一个表格（红色方框）。在图 4(b)-(e) 中，我们逐步增大图片和表格的尺寸。模型有效地调整了其他元素的位置和尺寸来适应这些变化。

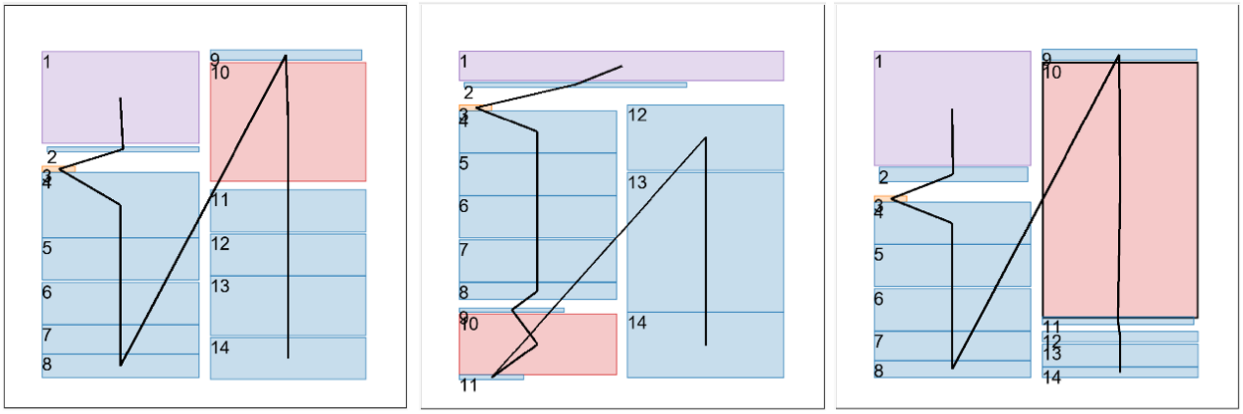


(a)初始布局 (b)放大图片 (c)放大图片 (d)放大表格 (e)放大表格

图 4. 体积修改示例。

5.3 长宽比修改

图 5说明了模型调整元素长宽比的能力。在图 5(a) 中展示了初始布局，包括图片和表格。图 5(b)-(c) 展示了对图片和表格长宽比的调整。模型展示了对元素尺寸和布局的适应性，调整了位置和大小以保持布局的协调。

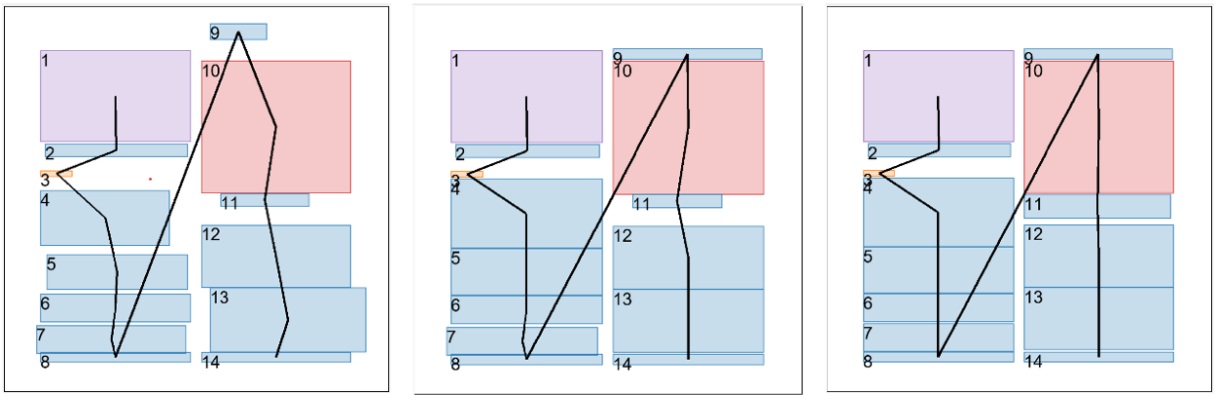


(a)初始布局　(b)加宽图片　(c)加高表格

图 5. 长宽比修改示例。

5.4 布局调整

如图 6所示，模型利用离群值检测功能对混乱的布局进行了逐步调整。从图 6(a) 的初始混乱布局开始，模型通过一系列的迭代，逐步调整元素位置，直到所有元素都整齐对齐。



(a)初始布局　(b)调整中途　(c)调整结束

图 6. 布局调整示例。

在所有实验中，模型都展现了卓越的能力，不仅在元素的单一属性调整上表现出色，而且在整体布局的优化上也表现出了高度的灵活性和适应性。这些结果验证了我们所提出方法的有效性和实用性。

6 总结与展望

6.1 总结

在本研究中，我们引入了一个新颖的基于 Transformer 的文档布局生成器，它精心设计以符合阅读规律和审美标准。我们的工作带来了几项显著的进步：

- 我们开发了一种算法来优化文档的阅读顺序，确保布局的逻辑性与直观性。
- 我们制定了一种方法来均衡地调整布局元素的体积和长宽比，从而提升了布局的多样性和视觉吸引力。
- 我们创造了一种布局采样策略，能够考虑到阅读顺序并有效避免空白或重叠，提高了布局的质量。
- 我们构建了一个用户友好的图形界面，使得模型的应用变得直观且容易操作。

此外，该模型展现了其广泛的适用性，它不仅适用于 PubLayNet 数据集，还有潜力在其他文档类型甚至非文档布局任务中发挥作用。

6.2 展望

尽管我们的生成器已经能够产出高品质的布局，但我们认识到仍有进步的空间。未来的工作可能会集中在以下几个方向：

- **元素个数控制：**现有模型对于元素数量的控制还过于刻板。我们期待开发出更加灵活的方式，以应对用户对于元素数量的不确定要求。
- **元素顺序控制：**同样地，模型对元素顺序的控制也显得过于精确。未来，我们希望模型能够处理更加模糊的顺序要求，以适应更广泛的设计布局。
- **元素体积和长宽比控制：**目前模型对体积和长宽比的控制是建立在软约束之上，有时可能与用户的指定出现偏差。我们希望未来的模型能够更准确地遵循这些参数，以减少可能的失真。

通过这些持续的努力，我们相信可以进一步完善我们的布局生成器，使其更好地服务于实际应用场景的需要。

参考文献

[1] David M Arroyo, Jan Postels, and Federico Tombari. Variational transformer networks for layout generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13642–13652, 2021.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[4] Kamal Gupta, Justin Lazarow, Alessandro Achille, et al. Layouttransformer: Layout generation and completion with selfattention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1004–1014, 2021.

[5] Anitha A Jyothi, Tom Durand, Ji He, et al. Layoutvae: Stochastic scene layout generation from a label set. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9895–9904, 2019.

[6] Xiang Kong, Liang Jiang, Hui Chang, et al. Blt: bidirectional layout transformer for controllable layout generation. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XVII*, pages 474–490. Springer Nature Switzerland, 2022.

[7] Hui-Yin Lee, Liang Jiang, Irfan Essa, et al. Neural design network: Graphic layout generation with constraints. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 491–506. Springer International Publishing, 2020.

[8] Jing Li, Jie Yang, Aaron Hertzmann, et al. Layoutgan: Generating graphic layouts with wire-frame discriminators. In *arXiv preprint arXiv:1901.06767*, 2019.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.