arXiv:2304.02173v1 [cs.CV] 5 Apr 2023

ChartReader：一种无需启发式规则的图表反渲染与理解统一框架

Zhi-Qi Cheng[1] Qi Dai[2] Siyao Li[1, 3] Jingdong Sun[1]Teruko Mitamura[1] Alexander G.Hauptmann[1]卡内基梅隆大学 [2]微软研究院 [3]亚马逊
{zhiqic,alex,teruko}@cs.cmu.edu, qid@microsoft.com, {siyaol,jingdons}@andrew.cmu.edu

摘要

图表是直观呈现复杂数据的有力工具，但由于图表类型多样、组件错综复杂，理解图表颇具挑战。现有的图表理解方法要么依赖启发式规则，要么过度依赖光学字符识别（*OCR*）系统，因而表现不尽理想。为解决这些问题，我们提出 *ChartReader*，一个将图表反渲染与理解任务无缝整合的统一框架。我们的方法包括一个基于 *Transformer* 的图表组件检测模块，以及一个扩展后的预训练视觉语言模型，用于图表到 *X*任务。通过从标注数据集中自动学习图表规则，我们的方法无需手动制定规则，从而减少工作量并提高准确性。我们还提出一种数据变量替换技术，并扩展预训练模型的输入嵌入和位置嵌入，以支持跨任务训练。我们在图表转表格、*ChartQA* 和图表转文本任务上评估了 *ChartReader*，结果表明其优于现有方法。我们提出的框架能够显著减少图表分析所需的人工工作，为构建通用图表理解模型迈出了一步。此外，我们的方法还支持与 *T5* 和 *TaPas* 等主流大语言模型即插即用地集成，从而扩展它们处理图表理解任务的能力。[1]

1. 引言

“图胜千言”这句格言凸显了各种网站和文章中的图表所蕴含的巨大价值，它们往往呈现了仅凭文字无法传达的知识。图表反渲染是指将图表转换为表格（即图表转表格 [11, 41, 42]），普遍被视为促进一系列下游任务的关键，例如图表问答（ChartQA [27, 45, 63]）和图表摘要——

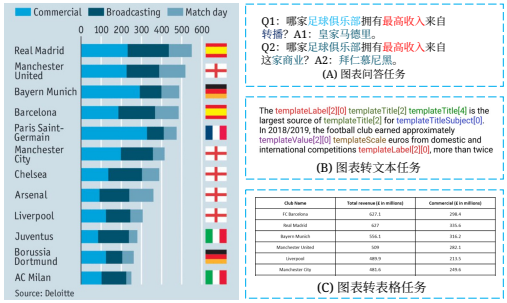


图1.图表反渲染和理解任务示意图。图表转表格任务旨在将图表转换为机器可读表格，而ChartQA和图表转文本任务分别涉及回答问题和总结图表内容。 [最好以彩色查看]。

摘要（图表转文本 [7, 15, 48]）。如图 1 所示，图表转表格任务旨在将图表识别为机器可读表格，而 ChartQA 和图表转文本任务则分别涉及回答预设问题和总结图表内容。早期研究也强调了这些研究任务之间的相互依赖性和相互价值 [22, 57, 59], ，突出了它们在图表理解研究中的关键作用。

尽管图表理解发挥着关键作用，现有研究却未能分别处理这三个子任务，更不用说提出一种通用解决方案了。如图 2 所示，图表有多种类型，每种类型都旨在传达领域特定知识，并且可能包含复杂的组件、纹理变化和斑点背景。面对这类复杂图表，现有的图表理解方法面临两个主要问题。

首先，现有的图表反渲染方法 [4, 11, 18,41, 42] （即图表转表格）依赖于启发式规则，这些规则需要大量领域知识和精力来制定。例如，ChartOCR [42], 作为一种开创性方法，需要先进行图表分类以识别类别，然后使用各种预定义的启发式规则检测不同的组件。为了避免复杂的规则制定，一些方法甚至只尝试一组有限的图表类型，例如

[1]代码位于 https://github.com/zhiqic/ChartReader

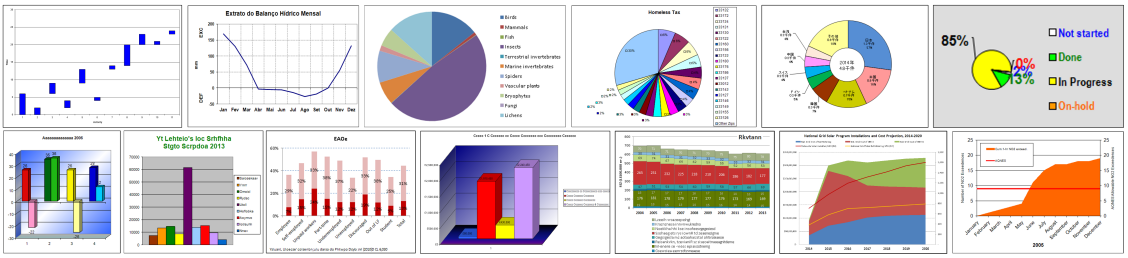


图 2. 展示了 EC400K 中图表的复杂性 [42], , 这些图表在类型、设计和视觉属性方面各不相同。图表可能包含复杂的组件、纹理变化和斑点背景，给图表理解带来挑战。 [彩色查看效果最佳]。

柱状图 [13, 50] 和折线图 [30, 43] 图表。这些局限性阻碍了从未知类别中提取数据的能力。因此，许多最新方法 [40, 45]甚至直接使用真实值中的表格来完成答案和摘要任务。显然，这些研究难以实现自动化，并且难以从真实世界的图表中提取数据。

另一方面，当前的图表理解方法，例如图表转文本 [7, 15, 48] 和 ChartQA [29,36, 44, 45, 53], , 往往严重依赖现成的光学字符识别系统，或从真实值中预先提取的表格。这些方法将图表反渲染视为黑箱，忽视了图表的视觉和结构信息，从而导致以下问题：(1) 图表转文本和图表问答任务退化为纯文本问答 [39, 44], , 因为它们无法从图表反渲染中提取视觉语义。这解释了为什么基于光学字符识别和端到端的方法，例如 LayoutLM [61], 、PresSTU [31], 、PaLI [8], 、CoCa [62],、Donut [32], 和 Dessurt [14], , 在图表理解方面表现不佳。(2) 图表转表格任务无法从图表理解任务中受益。由于缺乏对图表视觉语义的理解，现有系统（例如使用光学字符识别系统的系统 [39, 42], ）难以准确地将图表转换为表格。总之，我们认为，图表理解面临的问题源于过度依赖预定义规则，以及缺乏支持多任务处理的通用框架。

鉴于前面的分析，视觉语言模型似乎是构建统一框架的一个有前景的方向。然而，尽管 Pix2Struct [35], 这一种用于视觉情境语言的预训练策略已展现出优于基于OCR的模型的性能 [2, 5, 37,58], , 但它并不适用于图表反渲染。此外，尽管认识到了这一问题，Metcha [40] 由于缺乏更好的视觉语言模型，不得不依赖 Pix2Struct 作为骨干网络。尽管如此，Pix2Struct和 Metcha 都没有解决前面指出的两个问题：1）在表格反渲染中过度依赖启发式规则；2）尽管尝试纳入额外的图表理解任务，仍然严重依赖现有 OCR 系统。

为克服这些问题，我们提出了ChartReader，这是一种统一框架，可无缝整合图表逆渲染和理解任务。我们的方法包括一个无需规则的图表组件检测模块，以及一个 ex-

扩展的预训练视觉语言模型，用于图表到X（文本/表格/问答）任务。不同于基于启发式规则的方法，我们的方法采用基于Transformer的方法来检测图表组件的位置和类型，通过处理现有的带注释数据集，实现自动规则学习。为了增强跨任务训练，我们扩展了预训练模型的输入嵌入和位置嵌入，并引入了数据变量替换技术。具体而言，我们将图表到X（表格/文本）任务统一为问答问题，从而能够有效地解决多个图表理解任务。此外，模型生成数据变量而非实际值，以避免错误和幻觉，这提高了多任务训练中的一致性。我们的方法代表了迈向统一图表理解模型的一步，这一点已通过实验得到验证。所提出的框架有望减少图表分析所需的人工工作，为实现更高效、更准确的图表理解铺平道路。

总而言之，我们的贡献包括：1）一种统一的框架，无缝整合图表反渲染和图表理解任务；2）一个无需规则的图表组件检测模块，采用基于Transformer的方法，自动学习图表的规则；3）扩展预训练大语言模型的输入嵌入和位置嵌入，并采用数据变量替换技术来改进跨任务训练；4）通过实验验证我们的方法，证明其在图表理解任务中较现有方法取得了显著改进。

2. 相关工作

本节回顾了图表理解方面的相关工作，包括图表反渲染、问答和摘要生成，重点介绍其与以往工作的差异以及对视觉语言研究的影响。

图表反渲染，也称为图表转表格，涉及识别图表图像的组成组件，例如条形、饼图和图例，以提取图表所表示的底层数据。传统方法 [4, 18, 25, 49, 52] 依赖于基于边缘和颜色的人工设计规则，这些方法耗时较长，且难以推广到新的图表类型。基于深度学习的方法 [11, 41, 42] 利用目标检测和

文本识别网络已显示出能够以更好的泛化性准确检测图表组件。然而，近期一些工作 [39]尝试使用预训练的大规模语言模型（LLM）执行图表转表格任务，却未识别图表的结构和组件。尽管取得了进展，旗舰方法 ChartOCR [42], 仍严重依赖人工规则，并且针对每种图表类型使用不同的网络。相比之下，我们的方法消除了对启发式规则的需求，并且可以轻松扩展到未见过的图表样式。

图表问答（ChartQA）是一项利用视觉和文本信息回答与图表相关问题的任务。早期方法 [28] 依靠关系网络来表示图像对象之间，或视觉内容与问题之间的关系，同时引入动态编码 [26] 以处理词汇表外的候选答案。后来的方法 [27, 63]利用多模态模型，使用 LSTM [21] 和 DenseNet [23] 提取视觉和文本特征，并融合嵌入来回答问题。近期方法，如 OpenCQA [29], CRCT [36], ChartQA [44], PlotQA-M [45], 和 STL-CQA [53], , 在回答问题时引入了 Transformer，以捕捉复杂的视觉和文本信息。此外，一些方法 [40, 44]尝试使用预训练大语言模型对图表和语言数据进行建模，但它们通常忽略图表特征，而仅依赖真实标注表格。相比之下，我们的工作将图表摘要和反渲染任务统一到图表问答任务中，使用单一的序列到序列框架，将图表特征与不同任务的有意义方面优雅地结合到大语言模型中。

图表摘要，指的是图表转文本，旨在从视觉图表中生成自然语言摘要。传统方法 [17, 46, 51] 采用模板来提供对图表外观的简要描述，而其他方法 [16, 20]则使用启发式方法或基于规划的方法来创建多媒体演示。近年来，基于启发式规则的自然语言生成（NLG）技术已被应用，包括统计分析 [12, 54, 60]以推断洞见，以及编码器-解码器架构 [7, 15, 48] 以生成基于模板的图注。然而，这些方法存在一个共同局限，即依赖预定义的基于模板的方法，这可能导致其在语法风格和词汇选择方面缺乏通用性与多样性。相比之下，我们的研究提出了一种通用的序列到序列图表理解框架，该框架利用数据驱动模型生成更加多样且信息丰富的摘要。

视觉语言研究的进展。近年来，视觉语言研究主要聚焦于自然图像，依赖视觉 grounding 推理 [9, 38, 55] 和合成数据集 [1, 10, 24, 56] 进行评估。然而，这些研究并未涵盖.....的复杂性。现实世界的视觉语言，尤其是在图表理解任务中。尽管基于光学字符识别和端到端的方法（如LayoutLM [61], PresSTU [31], PaLI [8], CoCa [62],Donut [32], Dessurt [14], 和 Pix2Struct [35] ）已经针对视觉情境语言得到发展，但它们并未专门解决图表理解所带来的挑战。尽管编码器-解码器框架得到广泛应用，但它仍然需要特定的设计考量，例如确定哪些输入具有价值，以及消除人为规则。没有这些模块，现有模型（如 LaTr [5], GIT [58], DocFormer [2], 和 SelfDoc [37],）无法直接应用于图表理解。相比之下，我们的工作通过不要求预定义的启发式规则，并在编码器-解码器框架中统一数据提取与理解，消除了这些局限性，在多个数据集上取得了令人印象深刻的结果。

3. 统一图表阅读器框架

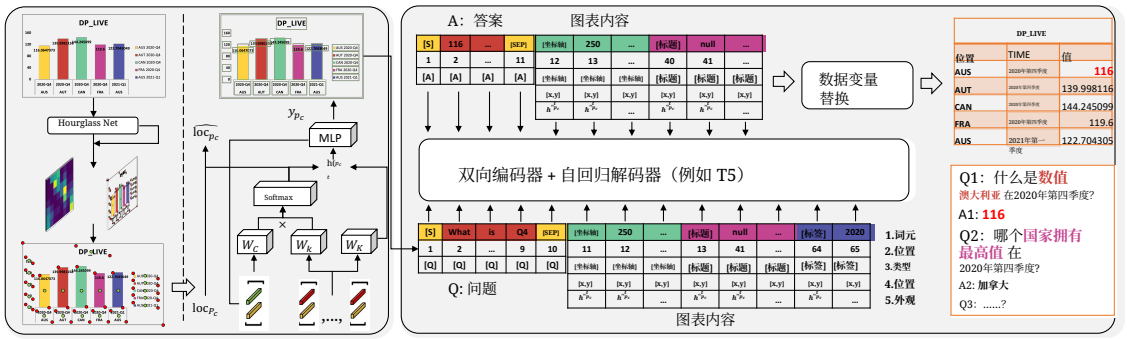
我们的统一框架旨在支持各种图表分析任务，包括图表转表格、图表转文本和 ChartQA。如图3所示，它由两个主要组件组成：（1）无需规则的图表组件检测模-块；以及（2）用于图表到X（文本/表格/问答）任务的扩展预训练视觉语言模型。

3.1. 图表组件检测

我们采用基于Transformer的方法来检测图表组件的位置和类型，而无需依赖启发式规则。我们的方法包含三个主要步骤：1）中心点/关键点检测，2）中心点与关键点分组，以及3）组件位置/类型预测。

克服启发式规则。其动机在于克服启发式规则方法在处理各种图表样式时的局限性，这些方法严重依赖领域特定知识和复杂的规则定义。通过处理已标注的现有数据集，我们的模型可以自动学习图表的规则。此外，优化后的框架可以无缝应用于其他下游图表理解任务。如图4所示，我们更新后的模型依赖从现有数据集推断出的中心点和关键点，而非启发式规则，来定位图表组件并克服风格变化的影响。我们将每个组件的左上角和右下角以及位置中心分别转换为中心点和关键点。尽管我们的方法仍需要大量标注数据，但我们无需为每种图表类型设计特定规则。我们的方法朝着统一图表理解模型迈出了一步，并已通过实验得到验证。

步骤-1：中心点/关键点检测。我们使用不含角点池化层的 Hourglass网络 [47]来检测图表组件的中心点$p_c \in \mathcal{C}$ 和关键点 $p_k \in \mathcal{K}$ ，以提高在不同图表样式之间的泛化能力。位置 $\widehat{\text{loc}}_p \in \mathbb{R}^2$ 和组件类型 $\hat{y}_p \in \{1, ..., \mathcal{T}\}$ 用于



(a) 图表组件检测　　(b) 图表反渲染与理解

图3展示了我们提出的用于各类图表分析任务的统一框架的组件，包括图表转表格、图表转文本和ChartQA。该框架由两个主要模块组成：（1）基于 Transformer 的图表组件检测模块，无需人工制定规则；（2）扩展的预训练视觉语言模型，可实现图表到X（文本/表格/问答）任务。这两个模块的结合使图表理解任务能够无缝集成并高效执行。

所有中心点和关键点均使用焦点损失L_{focal} [34] 和光滑 L_1 损失 [19]进行预测。我们对所有组件类型使用中心池化层，而不是针对每种组件类型采用不同的池化策略 [42], ，从而有效检测各种样式图表组件的中心点和关键点。

步骤2：中心点与关键点分组。对于每个检测到的中心点和关键点，我们提取其相应位置的特征，以获得初始嵌入。为了更好地确定图表组件，我们引入一个类型标记 ϕ_{p_c} 和 ϕ_{p_k} ，并使用多头注意力来获得中心点 p_c 和关键点 p_k 的权重，如下，

$$\mathcal{G}(p_c, p_k) = \left(\mathbf{W}_{\mathcal{C}}\left[h_{p_c}, \phi_{p_c}\right]\right)^T \mathbf{W}_{\mathcal{K}}\left[h_{p_k}, \phi_{p_k}\right], \quad (1)$$

其中 h_{p_c} 和 h_{p_k} 分别表示中心点和关键点的隐藏特征。矩阵$\mathbf{W}_{\mathcal{C}}$和$\mathbf{W}_{\mathcal{K}}$ 是中心点和关键点隐藏特征的投影矩阵。我们使用固定的正弦特征来编码绝对的x轴和y轴位置，并在相乘之前将 ϕ_{p_c} 和 ϕ_{p_k} 加到嵌入中。在使用公式1获得 $\mathcal{G}$ 的权重后，我们通过对整个关键点集合上的权重使用softmax函数进行归一化，从而计算最终的分组得分，如下：

$$\text{attn}(p_c, p_k) = \frac{\exp\left(\mathcal{G}(p_c, p_k)\right)}{\sum_{\bar{k}\in\mathcal{K}} \exp(\mathcal{G}(p_c, p_{\bar{k}}))}, \quad (2)$$

其中，softmax函数对来自同一组件的关键点 $k \in \mathcal{K}$ 进行排序，并为每个中心点 p_c筛选出最相关的那些（即 p_k）。该方法能够有效地将中心点和关键点归组到其对应的图表组件。

步骤 3：组件位置和类型预测。我们通过利用Hourglass网络的分组得分和关键点位置优化中心点位置，来预测每个图表组件的位置和类型。我们计算关键点位置的加权平均值。

$\widehat{\text{loc}}_{p_k}$ 对每个中心点 p_c 使用分组得分作为，

$$\widehat{\text{loc}}_{p_c} = \sum_{k\in\mathcal{K}} \text{attn}(p_c, p_k)\widehat{\text{loc}}_{p_k}. \quad (3)$$

为确保预测位置接近真实值位置 loc_{p_c}，我们使用位置损失 L_{loc}，其定义为，

$$L_{loc} = \sum_{c\in\mathcal{C}} |\widehat{\text{loc}}_{p_c} - \text{loc}_{p_c}|. \quad (4)$$

该方法监督优化过程，并有助于准确预测每个图表组件的位置。

为了预测图表组件的类型，我们计算关键点嵌入的加权和，并将分组得分作为，

$$\bar{h}_{p_c} = \sum_{k\in\mathcal{K}} \text{attn}(p_c, p_k)\mathbf{W}_{\mathcal{K}}[h_{p_k}, \phi_{p_k}]. \quad (5)$$

由此得到的中心点嵌入 h_{p_c} 随后经过一个多层感知机层，以获得所有组件类型上的预测概率分布，然后将其与真实值组件类型标签 y_{p_c} 使用交叉熵损失 L_{CLS} 进行比较，如下：

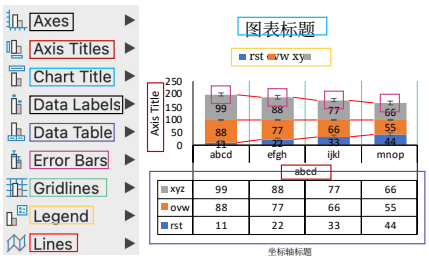
$$L_{\text{CLS}} = -\sum_{c\in\mathcal{C}} y_{p_c} \log(\text{softmax}(\bar{h}_{p_c})). \quad (6)$$

该过程监督优化过程，并有助于准确预测每个图表组件的类型。

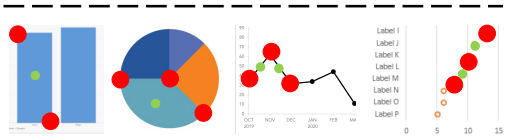
3.2. 图表反渲染与理解

我们提出了一个统一的图表理解框架，将Chart-to-X（文本/表格/问答）任务作为 ChartQA 任务处理，如图 X 所示。为改进跨任务训练，我们（1）扩展了预训练模型的输入嵌入和位置嵌入，并（2）采用了数据变量替换技术。

动机与推理。我们的方法主要基于两个原因：首先，我们将图表到X娱乐代理



(a) 字符组件示例



(b) 中心点和关键点转换示例

图4展示了将图表组件转换为中心点和关键点的过程，这些点根据现有数据集推断得出，并用于在更新后的模型中定位图表组件。这种方法无需启发式规则，使模型能够克服风格变化的影响。具体而言，我们分别将每个组件的左上角和右下角转换为中心点和关键点。 [彩色查看效果最佳]。

（表格/文本）任务作为问答问题。具体而言，对于图表转表格，坐标轴标签和图例的组合构成问题，而提取出的组件则作为答案。类似地，图表转文本被视为一种问答任务，用于填充通过删除冗余信息生成的空白模板。通过将这些此前相互独立的任务标准化为问答任务，我们可以有效地训练和解决多个图表理解任务。其次，这些任务的近期进展采用了序列到序列模型。为了在一个统一框架中处理所有这些任务，我们将预训练大语言模型扩展到多样化的图表理解任务。然而，整合这些任务颇具挑战，因此我们进行了大量实验，以确定如何扩展输入和位置编码。此外，我们提出了数据变量替换技术，以增强多任务训练的一致性。我们的研究结果为将预训练大语言模型扩展到图表理解提供了可能性，并且可以无缝扩展到 T5 和 TaPas 等主流大语言模型。

位置与输入嵌入. 为了支持图表转表格、图表转文本和 ChartQA 等任务，我们将输入嵌入和位置嵌入从自然语言扩展到了图表风格数据。融合嵌入 z_k 在序列的第 k个位置定义为，

$$z_k = \mathrm{LN}(z_k^{\mathrm{pos}} + (z_k^{\mathrm{token}} + z_k^{\mathrm{typ}} + z_k^{\mathrm{loc}} + z_k^{\mathrm{app}})), \quad (7)$$

其中LN()表示层归一化 [3]。首先，位置嵌入z_k^{pos} 被设为零，因为现代视觉-语言预训练模型使用相对位置嵌入。其次，输入嵌入包括图表组件的类型z_k^{typ} 、位置 z_k^{loc}和外观 z_k^{app} ，以及从数据集中其他文本信息获得的词元 z_k^{token}。

具体而言， z_k^{token}通过对文本词语的连接进行词元化生成，并表示为，

$$x^{\mathrm{token}} = \left\{ \begin{matrix} [\mathrm{S}], [\mathrm{w}_1], ..., [\mathrm{w}_\mathrm{m}], [\mathrm{SEP}], [\mathrm{y}_{\mathrm{p_c}}^1], w_{1,1} \\ , ..., w_{1,m}, [\mathrm{y}_{\mathrm{p_c}}^2], ..., [\mathrm{y}_{\mathrm{p_c}}^\mathrm{N}], w_{r_N,1}, ..., w_{r_N,m}, ..., \end{matrix} \right\},$$

其中， [S] 用于区分问题和答案，而 [SEP] 表示图表上下文的存在。为纳入有关图表组件的信息，引入特殊词元 $[\mathrm{y}_{\mathrm{p_c}}^\mathrm{i}]$ 来表示图表组件类型 $y_{p_c}^i \in \{1, ..., \mathcal{T}\}$（例如 [坐标轴]；见图4）。从每个图表组件中获取的其他文本词元表示为 $\{w_{1,1}, ..., w_{1,m}\}$。

为捕获有关图表的语义信息，我们引入了用于图表组件类型的可学习独热嵌入 z_k^{typ} 。图表中第k个词元的位置表示为 x_k^{loc}，它是基于相对边界框坐标的 4 维特征，如下所示：

$$x_k^{\mathrm{loc}} = (x_k^{\mathrm{min}}/W_{\mathrm{im}}, y_k^{\mathrm{min}}/H_{\mathrm{im}}, x_k^{\mathrm{max}}/W_{\mathrm{im}}, y_k^{\mathrm{max}}/H_{\mathrm{im}}),$$

其中，$(x_k^{\mathrm{min}}, y_k^{\mathrm{min}})$和 $(x_k^{\mathrm{max}}, y_k^{\mathrm{max}})$分别表示第k 个词元边界框的左上角和右下角坐标，而 W_{im} 和 H_{im} 分别表示图表的宽度和高度。为考虑每个图表组件的外观，我们拼接中心点及其对应关键点的特征，以获得最终的外观嵌入 z_k^{app} 。

数据变量替换技术。我们通过引入数据变量替换技术来改进训练过程。在训练期间，模型生成相应的数据变量，而不是实际值，从而避免因将数据记录视为普通词元而导致的错误和幻觉。例如，表格单元格中的数值被替换为“product1”“product2”等数据变量。这种方法提高了生成的摘要、表格和答案的准确性与事实一致性，尤其是在涉及多个数据记录时。

为监督数据变量的使用，我们引入了一个新的损失项，用于惩罚模型生成与任何数据变量都不匹配的词元。该损失项使用函数 $D(x)$：如果词元 x 匹配某个数据变量，则返回该数据变量，否则返回 null。损失项定义如下：

$$L_{\mathrm{var}} = -\frac{1}{T}\sum_{t}^{T}\sum_{i}^{N_t}\sum_{j \in V} I_{D(x_i)=v_j} \log P_{i,j}(t), \quad (8)$$

其中， T 是生成输出的长度， N_t 是第 t个输出中的词元数量，并且$P_{i,j}(t)$是在第 t个时间步将第 i个词元生成为第 j个数据变量的概率。$I_{D(x_i)=v_j}$ 是一个指示函数：如果 $D(x_i) = v_j$，则其值为1，否则为0。

最终的优化目标是优化两个损失之和，其中 α 作为平衡两个损失的超参数，

$$L = L_{\text{ans}} + \alpha L_{\text{var}}, \tag{9}$$

其中， α 是用于平衡两项损失的超参数。损失 L_{ans} 可能会根据具体的图表理解任务进行略微调整。总之，我们采用数据变量替换的方法，可以同时支持图表转文本、图表转表格和ChartQA任务。所引入的损失项还确保了在训练过程中正确使用数据变量。

表 1. 用于多种图表理解任务的数据集，包括图表转表格、ChartQA 和图表转文本。

任务	数据集	#图表	#QAPs
图表转文本	C2T [48]	83K	-
图表转表格	EC400K [42]	387K	-
ChartQA	FQA [28]	180K	2.4M
ChartQA	PlotQA [45]	224K	28M
ChartQA	ChartQA [44]	22K	33K
ChartQA	DVQA [26]	300K	3.5M

4. 实验

4.1. 评估任务和数据集

我们在多项图表理解任务上进行了实验，包括图表转表格、ChartQA 和图表转文本，如表 1 所示。

图表转文本任务。对于图表转文本任务，我们使用了 C2T [48] 数据集，该数据集包含两个子集：皮尤和 Statista。皮尤子集由来自皮尤研究中心的图表图像及自动提取的摘要组成，而 Statista 子集则由来自 Statista 的图表图像及人工撰写的摘要组成。该数据集提供了多样的图表样式和文本摘要，有助于开发高效的图表转文本模型。

图表转表格任务。为了评估我们方法在图表转表格任务中的有效性，我们使用了 EC400K 数据集 [42]。该数据集包含来自公开 Excel工作表的386,966 张真实世界图表图像，并提供图表的边界框位置和数值读数。EC400K 数据集提供了种类繁多的图表类型和样式，超越了以往用于图表理解研究的数据集。这使我们能够在多样且具有挑战性的真实世界图表数据上验证其性能。

图表问答任务。我们在四个数据集上评估了用于 ChartQA的方法：FQA [28], DVQA [26], PlotQA [45], 和 ChartQA [44]。FQA 数据集包含带有二元答案问题模板的图表，并拥有两个验证集和两个不公开的测试集。DVQA 是一个合成数据集，提供视觉元素及其元数据的精确位置和外观，其中包括两个测试任务：熟悉测试集和新颖测试集。PlotQA 是一个大型且用于图表理解任务的公开可用数据集，包含根据现实世界数据生成的图表。它提供了两个基准测试集，即 PlotQA-V1 和 PlotQA-V2，后者将前者作为子集包含在内。ChartQA是一个较新的开放域图表问答数据集，在两个子集上进行评估：增强集和人工集，其中增强集由机器生成，具有更强的抽取性，而人工集由人工编写，需要更复杂的推理。

评估指标。我们在研究中采用了针对任务的指标。图表转表格任务采用了 ChartOCR[42] 中使用的指标，以确保公平性。对于 ChartQA，我们使用了标准准确率，并采用宽松的正确性标- 准 [44, 45]，允许最大为% 的数值误- 差。此外，我们使用 BLEU4 来评估图表转文本任务。

4.2. 不同任务的训练细节

在本节中，我们概述了我们的方法在不同任务上的训练细节。图表组件检测模型在 8×台A6000GPU 上进行训练，而所有其他基于预训练大语言模型的实验均使用 64 个 GCP-TPUv3 进行微调。

图表转文本任务。我们以 2e-5 的学习率、16 的批次大小以及 512 的最大序列长度，对模型进行了 10,000 步的微调。

图表转表格任务。我们在 EC400K 数据集[42], 上训练了图表组件检测模块，该数据集目前是用于图表转表格任务的最大数据集。中心点和关键点位置的真实标注由原始数据集中标注的边界框推导得出。训练期间，我们使用 Adam 优化器[33]，学习率为 2.5e-4，并在最后 5,000 个批次中将学习率降至 2.5e-5，批次大小为 32。我们使用 Soft-NMS[6] 合并热力图中的关键点。超参数根据验证集进行设置，并在端到端训练中采用早停。尽管我们没有针对其他任务微调图表组件检测模块，但在其他图表理解任务上训练得到的最终模型仍可用于提取图表转表格任务中的表格数据。其余设置与先前的工作 ChartOCR [42]相同。

图表问答任务。我们在不同数据集上对图表问答任务进行了微调。对于 FigureQA，我们使用二元交叉熵损失和 Adam优化器，基础学习率为5e-4。学习率从第 15 个周期衰减至第25个周期，衰减因子为0.7。对于DVQA，我们使用多项交叉熵损失和 Adam优化器，基础学习率为7e-4。学习率从第 15 个周期衰减至第25个周期，衰减因子为 0.6。对于 PlotQA，我们使用二元交叉熵损失和 Adam 优化器，基础学习率为5e-4。通过随机为问题分配错误答案来生成负例，并使用线性学习率调度器训练模型 20 个周期。

表 2. 不同方法在图表问答任务上的性能比较。表中展示了我们提出的模型 TaPas（我们的方法）和 T5（我们的方法）在四个数据集 FQA、DVQA、PlotQA 和 ChartQA 上的结果。基于 T5 的模型在所有数据集上均取得了最先进的性能，大幅优于此前的方法。宽松正确性标准允许的数值误差最大为 5%。对于 DVQA 数据集，TF 和 TN 分别表示熟悉测试集和新颖测试集。

	FQA				DVQA		PQA		ChartQA	
方法	**Val1**	**val2**	**测试集 1**	**测试集2**	**TF**	**TN**	**测试集 V1**	**测试 V2**	**Val**	**Test**
图像+问题 [26]	59.41	57.14	-	56.04	21.06	21.00	-	-	-	-
PReFIL [27]	94.84	93.26	**94.88**	93.16	69.66	69.53	57.91	10.37	4.53	4.80
CRCT [36]	94.61	85.04	94.23	84.77	-	-	76.94	34.44	-	-
PlotQA [45]	-	-	-	-	57.99	59.54	53.96	22.52	36.15	38.00
TaPas [44]	90.32	90.43	89.52	89.57	48.82	48.68	15.09	12.90	39.68	41.28
V-TaPas [44]	91.46	91.45	90.68	90.64	94.43	94.54	65.30	42.50	42.60	45.52
T5 [44]	87.97	87.83	87.56	87.57	89.01	89.01	72.62	56.22	40.15	41.04
VL-T5 [44]	88.60	88.49	88.20	88.18	93.75	93.75	75.90	56.02	38.43	41.56
TaPas（本文）	91.12	91.40	91.15	91.25	92.20	94.30	74.20	56.20	48.30	50.20
T5（本文）	**95.50**	**95.80**	94.40	**93.40**	**95.40**	**96.50**	**78.10**	**59.30**	**49.50**	**52.60**

4.3. 最先进方法对比

图表转文本任务的结果。表 3 展示了图表转文本任务的结果，其中我们将基于 T5 的模型与先前最先进的方法 PaLI-17B（分辨率 224）和 Pix2Struct 在两个数据集 Pew 和 Statista 上的性能进行了比较。我们的基于 T5 的模型在两个数据集上均取得了最佳性能，显著优于先前的方法。通过将我们的模型与原始 T5 模型进行比较，我们认为，性能优势源于模型能够生成多样且信息丰富的摘要。值得注意的是，尽管模型规模较小，我们基于 T5 的模型在两个数据集上的表现均优于 PaLI-17B （分辨率 588 ）。这表明，即使模型规模较小，我们的方法也能取得良好的性能。

表 3. 图表转文本任务在皮尤和 Statista 数据集上的结果。我们的 T5 模型在两个数据集上都取得了最先进的性能，超越了此前最先进的方法。

方法	**Pew**	**Statista**
T5 [40]	10.5	35.3
PaLI-17B（分辨率 224） [40]	10.0	40.2
PaLI-17B（分辨率 588） [40]	11.2	41.4
Pix2Struct [35]	10.3	38.0
Matcha [40]	12.2	39.4
T5（我们的）	**14.2**	**44.2**

图表转表格任务的结果。在表 4中，我们报告了所提出方法在 EC400K数据集 [42]上的图表转表格任务结果。我们的方法取得了最先进的性能，超越了RotationRNN和Faster-RCNN 等以往方法；这些方法仅依赖图像分类和目标检测。该比较突显了我们关键点检测方法的优越性-

表 4. Chart-to-Table 任务中，ChartReader 与先前最先进方法在 EC400K 上的比较。报告的 GPU 小时数指在 8 × A6000 GPU 上进行评估所用的 GPU 总处理时间，大致相当于运行时间的 8 倍，其中包括 IO 和数据预处理时间。粗体表示最佳性能。

方法	**Bar**	**Pie**	**Line**	**GPU 小时**
Revision [52]	0.58	0.84	-	-
Faster-RCNN [41, 11]	0.80	-	-	-
Rotation-RNN [41]	-	0.80	-	-
ChartOCR [42]	0.92	0.92	0.96	57h
Ours	**0.95**	**0.95**	**0.97**	22h

相比边界框检测的方法。本方法也优于 Revision 和 Char-tOCR 等早期尝试，这些方法高度依赖手工设计的启发式规则。值得注意的是，ChartOCR 代表了当前图表转表格任务的最先进水平。本方法性能更优可归因于两个因素。首先，我们消除了对启发式规则的需求，通过对中心点和关键点进行分组来学习识别图表组件。其次，识别出的图表组件会进一步用于后续的图表转表格和图表问答任务，使模型能够更好地理解图表的结构和语义信息，从而提升数值评估性能。此外，如表 2 所示，与 ChartOCR 相比，本方法仅需显著更少的 GPU小时即可达到这一优异性能。具体而言，本方法仅需 22 GPU小时，而 ChartOCR 使用了 57 GPU小时。这表明，本方法不仅优于以往方法，而且计算资源利用效率也更高。

图表问答任务的结果。表 2展示了我们提出的模型 TaPas（Ours）和 T5（Ours）在

表5.图表组件检测的消融实验，比较完整模型与两个变体的性能，这两个变体分别不具备CT点检测或组检测。完整模型在全部三种图表类型上均优于消融模型，凸显了这两个组件对于准确的图表组件检测的重要性。

方法	Bar	Pie	Line
不含 CT Point	0.83	0.73	0.63
不含组	0.77	0.81	0.52
Ours	0.95	0.95	0.97

表 6。针对PlotQA-V1 和 PlotQA-V2 数据集的输入编码消融实验。表中显示，完整模型的性能优于所有消融版本，而位置和外观嵌入对整体性能的贡献最大。

消融标记	PlotQA-V1	PlotQA-V2
无类型	60.20	48.20
w/o 位置	64.20	51.20
无外观	59.10	45.20
不含 CCD	52.30	40.20
我们的方法（TaPas）	74.20	56.20

ChartQA 问答任务中的 ALL 数据集。我们的方法利用图表的结构和语义信息来回-答问题，因此非常适合 ChartQA 等复杂的图表理解任务。具体而言，我们基于 T5 的模型在 FQA 数据集上的表现以微小优势超过了此前最先进的方法 PReFIL，并在其他三个数据集上取得了最先进的性能。我们的模型受益于图表理解和提取任务上的联合训练，从而能够更好地理解图表的语义和结构信息。我们的基于 T5的模型在DVQA 和 PlotQA 数据集上取得了最先进的性-能，超过了此前的方法，包括当前最先进的方法V-TaPas，并且在两个测试集上均表现优于后者。此外，我们基于T5 的模型在 ChartQA数据集上取得了最先进的性能，以较大优势超过了此前的方法，表明其对未见数据具有良好的泛化能力。

表 7. 超参数 α 在数据变量替换技术中的影响。

	FQA	DVQA	PlotQA	图表问答	C2T
α	0.3	0.2	0.6	0.7	0.5
Rate	5%	5%	20%	40%	30%

5. 消融实验

图表组件检测。我们将完整模型与两个分别不包含CT点检测或组检测的变体进行了比较。如表5所示，完整模型在全部三种图表类型上均优于消融模型，而CT点模块提升了折线图上的性能，

表 8。预训练数据集对图表问答任务影响的消融实验结果。

模型	PT	EF	MT	Val	Test
T5	C4+(PQA)	CQA	✗	38.4	39.2
T5	C4+(PQA)	CQA	✔	42.3	41.5
T5	C4+(PQA+CQA)	CQA	✗	39.2	39.5
T5	C4+(PQA+CQA)	CQA	✔	45.5	43.2
T5	C4+(PQA+CQA+C2T)	CQA	✗	41.2	42.2
T5	C4+(PQA+CQA+C2T)	CQA	✔	49.5	52.6

而分组模块对于条形图更为有效。这些结果凸显了两个组件对于准确进行图表组件检测的重要性。

输入编码。我们对输入编码方法进行了消融实验，比较了完整模型和不同消融版本在 PlotQA-V1 和 PlotQA-V2 数据集上的表现。表 6 显示，完整模型显著优于所有消融版本，其中位置和外观嵌入对整体性能的贡献最大。这表明，空间和视觉信息对于图表理解任务至关重要。

数据变量替换中的超参数 α 。本节通过在五个数据集上进行消融实验，考察数据变量替换技术中超参数 α 的影响。如表 7 所示， α 的最优值因数据集而异，范围为 0.2 至 0.7。我们推测，更复杂的任务需要为替换后的数据变量赋予更高权重。实验结果证明了数据变量替换技术的有效性。

多任务预训练。我们通过消融实验评估所提出的图表组件检测和多任务训练方法在图表问答任务中的有效性。如表 8 所示，使用包含三种图表相关任务的 C4+ 预训练数据集训练的模型，在验证集和测试集上均取得了最佳性能。这些结果表明，采用多任务训练至关重要。

6. 结论

在本文中，我们提出了 ChartReader，这是一种统一的图表理解框架，可无缝整合图表反渲染与图表理解任务。我们的方法采用基于 Transformer 的图表组件检测模块，以及扩展的预训练视觉语言模型来执行图表理解任务，无需手动制定规则，同时提高了准确性。通过大量实验，我们证明，ChartReader 在图表转表格、ChartQA 和图表转文本任务中的表现优于现有方法。我们的框架有望显著减少图表分析中的人工工作，并推动通用图表理解模型的发展。

补充材料

关于超参数的讨论 α。超参数 α 对于实现预训练模型的高性能和训练效率至关重要。它用于数据变量替换技术，以控制模型如何处理输入数据中的未知标记。我们进行了额外实验，以研究 alpha参数的影响，如图 A1所示。实验结果表明，随着开放式问题数量的增加， α 参数的权重也随之增加，因为模型需要更频繁地使用数据变量替换技术，以避免随机猜测未知标记所导致的错误。然而，为 α 参数选择不恰当的值可能会使模型难以收敛。因此，选择alpha 参数的值时需要慎重考虑，以最大限度地提高模型的性能和训练效率。

关于训练步骤的讨论。在表 2中，我们分析了训练问答任务或文本任务的最佳序列，并发现最后训练问答任务能够带来更好的性能，尤其是在包含问答数据集时。同样，我们观察到最后训练图表转文本任务也能获得更好的性能。我们还探究了问答任务中各个数据集的训练顺序，发现从易到难进行训练能够取得更好的结果。然而，我们发现多任务训练，即将 ALL 数据集合并训练，能够获得更好的性能，这可能是由于问答任务与文本任务之间固有的互补性。此外，我们还考虑了不同数据集的规模，并提供了融合权重，该权重显示在表 1 的“比率”中。

不同模型变体的比较。我们提出的方法提升了不同模型的性能，这一点从我们对 T5 模型与T5S 等其他模型变体的比较中可以看出。与原始T5基线相比，结合 CCD 模块和 MT显著提高了模型的泛化性能和稳定性，如表 3 所示。然而，由于额外的训练时间和输入/输出操作，纳入 CCD 模块会略微降低训练效率。尽管如此，优化这些操作不会显著增加训练时间，且TPU 小时数的增加仍处于可接受范围内。我们将继续致力于该领域的研究，并进一步优化所提出的方法，以提升其有效性。

局限性讨论。虽然我们提出的方法在处理多项任务以及支持跨数据集和跨任务学习方面取得了令人鼓舞的结果，但模型的性能可能会受到预训练数据集的规模和质量以及具体微调任务的影响。选择合适的超参数也颇具挑战，可能需要进行大量实验。这些局限性表明，仍需进一步研究以优化-化我们提出的方法并提高其性能。尽管如此，我们的工作为通用图表理解框架的开发提供了有价值的见解。

表 1。数据变量替换技术中超参数 α 的影响。QA 和文本任务被划分为难度级别不同的子数据集，“比率”列显示用于多任务训练的融合权重。

	FQA	DVQA	PlotQA	图表问答	C2T
α	0.3	0.2	0.6	0.7	0.5
Rate	5%	5%	20%	40%	30%

表 2. 预训练数据集和训练步骤。本表展示了组合预训练数据集的详细信息，以及不同任务（包括问答任务、文本任务和图表转文本任务）所涉及的训练步骤。本表还包括用于组合问答任务中不同数据集的融合权重。

	ChartQA		图表转文本	
实验设置	**Val**	**Test**	**Pew**	**Statista**
问答->文本	46.2	47.2	13.4	43.2
文本->问答	48.2	51.6	12.7	41.2
按难度划分的问答	48.3	51.2	-	-
随机顺序的问答	47.5	50.4	-	-
混合训练	**49.5**	**52.6**	**14.2**	**44.2**

致谢

本研究获得了美国空军研究实验室根据协议编号 FA8750-19-2-0200 提供的支持；获得了通过GAILA项目资助的美国国防高级研究计划局（DARPA）拨款（授予编号HR00111990063）；以及获得了通过 AIDA项目资助的美国国防高级研究计划局（DARPA）拨款（ FA8750-18-20018）。

美国政府有权为政府目的复制和分发再版材料，尽管其上有任何版权声明。本文所述观点和结论属于作者，不应被解释为必然代表美国空军研究实验室或美国政府明示或默示的官方政策或认可。

参考文献

[1] Jacob Andreas、Marcus Rohrbach、Trevor Darrell 和Dan Klein。神经模块网络。载于 *IEEE*计算机视觉与模式识别会议论文集，第39–48页，2016年。[2] Srikar Appalaraju、Bhavan Jasani、Bhargava Urala Kota、Yusheng Xie和 R Manmatha。DocFormer：用于文档理解的端到端 Transformer。载于 *IEEE/CVF*国际计算机视觉会议论文集，第993–1003页，2021年。

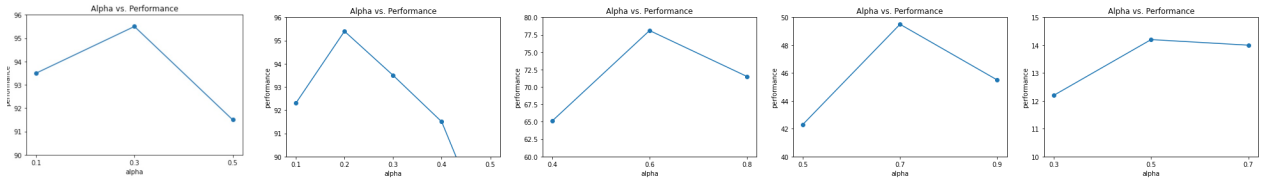


图 A1 展示了超参数 α 对模型在图表理解方面性能的影响。实验结果表明，随着开放式问题数量的增加， α 参数的权重也随之增加，因为模型需要更频繁地使用变量替换技术，以避免因随机猜测未知标记而导致的错误。适当选择 α 参数对于实现预训练模型的高性能和训练效率至关重要。

表 3。不同模型变体之间的性能比较。结果表明，与原始T5基线相比，引入 CCD 模块和 MT 显著提升了模型的泛化性能和稳定性。所提出的方法提升了不同模型的性能，这一点可从 T5模型与其他变体（如 T5S）的比较中看出。该表还显示了训练时间和 TPU 小时数的增加。

					PQA（PlotQA）		CQA (ChartQA)		C2T（表格转文本）		
模型	**PT**	**EF**	**CCD.**	**MT.**	**测试集V1**	**测试 V2**	**Val**	**Test**	**Pew**	**Statista**	**TPU 小时**
T5	C4	SQuAD	✗	✗	31.3	21.2	16.9	18.0	6.9	22.8	-
T5	C4+(CQA)	PQA	✗	✗	65.7	42.9	23.4	22.3	-	-	17
T5	C4+(CQA)	PQA	✔	✗	68.4	51.7	26.4	24.2	-	-	31
T5	C4+(CQA)	PQA	✗	✔	69.5	52.5	28.3	27.2	-	-	21
T5	C4+(CQA)	PQA	✔	✔	73.5	56.5	32.6	31.6	-	-	37
T5	C4+(PQA)	CQA	✗	✗	45.6	24.8	38.4	39.2	-	-	13
T5	C4+(PQA)	CQA	✔	✗	53.9	36.6	38.7	39.6	-	-	28
T5	C4+(PQA)	CQA	✗	✔	56.8	40.8	37.6	36.6	-	-	19
T5	C4+(PQA)	CQA	✔	✔	63.8	47.6	42.3	41.5	-	-	35
T5	C4+(CQA+C2T)	PQA	✗	✗	31.8	21.8	27.4	26.2	-	-	24
T5	C4+(CQA+C2T)	PQA	✔	✗	68.6	45.9	30.3	36.4	-	-	36
T5	C4+(CQA+C2T)	PQA	✗	✔	70.5	53.5	29.4	38.4	-	-	27
T5	C4+(CQA+C2T)	PQA	✔	✔	74.6	55.2	34.2	31.5	-	-	41
T5	C4+(PQA+C2T)	CQA	✗	✗	48.6	18.8	38.3	36.4	-	-	22
T5	C4+(PQA+C2T)	CQA	✔	✗	52.8	35.6	27.3	40.3	-	-	35
T5	C4+(PQA+C2T)	CQA	✗	✔	54.6	36.9	38.5	39.2	-	-	25
T5	C4+(PQA+C2T)	CQA	✔	✔	65.7	49.7	45.5	44.3	-	-	38
T5	C4+(ALL)	PQA	✗	✗	61.2	43.4	29.3	31.5	-	-	28
T5	C4+(ALL)	PQA	✔	✗	72.2	52.5	36.2	34.4	-	-	41
T5	C4+(ALL)	PQA	✗	✔	73.3	54.2	35.4	33.3	-	-	32
T5	C4+(ALL)	PQA	✔	✔	**78.1**	**59.3**	39.4	37.4	-	-	47
T5	C4+(ALL)	CQA	✗	✗	43.8	15.7	41.2	42.2	-	-	26
T5	C4+(ALL)	CQA	✔	✗	48.1	31.6	44.4	46.5	-	-	38
T5	C4+(ALL)	CQA	✗	✔	52.7	34.8	45.4	45.4	-	-	27
T5	C4+(ALL)	CQA	✔	✔	72.7	53.5	**49.5**	**52.6**	-	-	44
T5	C4+(ALL)	C2T	✗	✗	-	-	-	-	9.6	34.7	27
T5	C4+(ALL)	C2T	✔	✗	-	-	-	-	12.5	37.5	42
T5	C4+(ALL)	C2T	✗	✔	-	-	-	-	13.2	38.8	34
T5	C4+(ALL)	C2T	✔	✔	-	-	-	-	**14.2**	**44.2**	44
TaPas	C4+(ALL)	PQA	✔	✔	74.4	56.5	37.5	36.5	-	-	54
TaPas	C4+(ALL)	CQA	✔	✔	48.7	50.6	48.8	50.7	-	-	47
Tapas	C4+(ALL)	C2T	✔	✔	-	-	-	-	12.8	38.8	43
T5S	C4+(ALL)	PQA	✔	✔	72.3	55.3	34.5	32.4	-	-	34
T5S	C4+(ALL)	CQA	✔	✔	64.3	47.3	44.4	43.4	-	-	32
T5S	C4+(ALL)	C2T	✔	✔	-	-	-	-	12.4	36.7	31

[3] Jimmy Lei Ba、Jamie Ryan Kiros 和 Geoffrey E Hin- ton。层归一化。*arXiv* 预印本 *arXiv:1607 06450*，2016。。[4] Abhijit Balaji、Thuvaarakkesh Ramanathan 和 Venkatesh- warlu Sonathi。图表文本：全自动图表图像描述符。*arXiv* 预印本 *arXiv:1812.10636*，2018。[5] Ali Furkan Biten、Ron Litman、Yusheng Xie、Srikar Ap- palaraju 和 R Manmatha。LATr：用于场景文本视觉问答的布局感知型 Transformer。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第16548–16558页，2022年。[6] Navaneeth Bodla、Bharat Singh、Rama Chellappa 和 Larry S Davis。Soft-NMS——仅用一行代码改进目标检测。载于*IEEE*国际计算机视觉会议论文集，第5561–5569页，2017年。[7] Charles Chen、Ruiyi Zhang、Eunyee Koh、Sungchul Kim、Scott Cohen、Tong Yu、Ryan Rossi 和 Razvan Bunescu。通过推理和序列级训练生成图示说明。*arXiv* 预印本 *arXiv:1906.02850*，2019年。[8] Xi Chen、Xiao Wang、Soravit Changpinyo、AJ Piergiovanni、Piotr Padlewski、Daniel Salz、Sebastian Goodman、Adam Grycner、Basil Mustafa、Lucas Beyer 等。PaLI：联合扩展的多语言视觉语言模型。*arXiv* 预印本 *arXiv:2209.06794*，2022年。[9] Zhi-Qi Cheng、Qi Dai、Siyao Li、Teruko Mitamura 和 Alexander Hauptmann。GSRFormer：采用交替语义注意力精炼的基于视觉依据的情境识别 Transformer。载于*ACM*多媒体国际会议论文集，第3272–3281页，2022年。[10] Zhi-Qi Cheng、Xiao Wu、Siyu Huang、Jun-Xiu Li、Alexander G Hauptmann 和 Qiang Peng。学习迁移：通过多任务神经模型搜索实现可泛化属性学习。载于第*26*届*ACM*多媒体国际会议论文集，第90–98页，2018年。[11] Jinho Choi、Sanghun Jung、Deok Gun Park、Jaegul Choo 和 Niklas Elmqvist。为非视觉者进行可视化：让视障人士也能使用可视化。载于计算机图形论坛，第38卷，第249–260页。Wiley Online Library，2019年。[12] Zhe Cui、Sriram Karthik Badam、M Adil Yalcin 和 Niklas Elmqvist。DataSite：通过计算基于洞察的推荐，主动进行可视化数据探索。《信息可视化》，18(2):251–267，2019年。[13] Siri Chandana Daggubati、Jaya Sreevalsan-Nair 和 Komal Dadhich。BarChartAnalyzer：从图像中提取和总结条形图数据。《*SN*计算机科学》，3(6):1–19，2022年。[14] Brian Davis、Bryan Morse、Brian Price、Chris Tensmeyer、Curtis Wigington 和 Vlad Morariu。使用Dessurt进行端到端文档识别与理解。载于计算机视觉——*ECCV 2022*研讨会：以色列特拉维夫，*2022*年*10*月*23*日至*27*日，论文集，第四部分，第280–296页。Springer，2023年。[15] Seniz Demir、Sandra Carberry 和 Kathleen F McCoy。以文本形式总结信息图形。计算语言学，38(3):527–574，2012年。[16] Massimo Fasciano 和 Guy Lapalme。从表格数据协调生成图形与文本的意图。知识与信息系统，2(3):310–339，2000。[17] Leo Ferres、Gitte Lindgaard、Livia Sumegi 和 Bruce Tsuji。评估一种用于提升图表与图形可访问性的工具。*ACM*《计算机-人类交互汇刊》（*TOCHI*），20(5):1–32，2013。[18] Jinglun Gao、Yin Zhou 和 Kenneth E Barner。VIEW：用于提升图表图像可访问性的视觉信息提取组件。载于*IEEE*国际图像处理会议论文集，第2865–2868页。IEEE，2012年。[19] Ross Girshick。Fast R-CNN。载于*IEEE*国际计算机视觉会议论文集，第1440–1448页，2015年。[20] Nancy L Green、Giuseppe Carenini、Stephan Kerpedjiev、Joe Mattis、Johanna D Moore 和 Steven F Roth。AutoBrief：一种用于自动生成整合文本与信息图形的简报的实验系统。国际人机交互研究期刊，61(1)：70–132，2004年。[21] Sepp Hochreiter 和 J¨ urgen Schmidhuber。长短期记忆。《神经计算》，9(8):1735–1780，1997。[22] Danqing Huang、Jinpeng Wang、Guoxin Wang 和 Chin-Yew Lin。从图表图像中提取视觉风格以重新设计图表。载于国际模式识别会议论文集，第7625–7632页。IEEE，2021年。[23] Gao Huang、Zhuang Liu、Laurens Van Der Maaten 和 Kilian Q Weinberger。密集连接卷积网络。载于*IEEE*计算机视觉与模式识别会议论文集，第4700–4708页，2017年。[24] Justin Johnson、Bharath Hariharan、Laurens Van Der Maaten、Li Fei-Fei、C Lawrence Zitnick 和 Ross Girshick。CLEVR：用于组合语言和基础视觉推理的诊断数据集。载于*IEEE*计算机视觉与模式识别会议论文集，第2901–2910页，2017年。[25] Daekyoung Jung、Wonjae Kim、Hyunjoo Song、Jeong-in Hwang、Bongshin Lee、Bohyoung Kim 和 Jinwook Seo。ChartSense：从图表图像中交互式提取数据。载于*CHI*计算系统中的人为因素会议论文集，第6706–6717页，2017年。[26] Kushal Kafle、Brian Price、Scott Cohen 和 Christopher Kanan。DVQA：通过问答理解数据可视化。载于*IEEE/CVF*计算机视觉与模式识别会议论文集，第5648–5656页，2018年。[27] Kushal Kafle、Robik Shrestha、Scott Cohen、Brian Price 和 Christopher Kanan。使用高效双模态融合回答有关数据可视化的问题。载于*IEEE/CVF*冬季计算机视觉应用会议论文集，第1498–1507页，2020年。[28] Samira Ebrahimi Kahou、Vincent Michalski、Adam Atkinson、´Akos K´ad´ar、Adam Trischler 和 Yoshua Bengio。FigureQA：用于视觉推理的带注释的图示数据集。*arXiv*预印本 *arXiv:1710.07300*，2017年。[29] Shankar Kantharaj、Xuan Long Do、Rixie Tiffany Ko Leong、Jia Qing Tan、Enamul Hoque 和 Shafiq Joty。OpenCQA：

基于图表的开放式问答。*arXiv*预印本 *arXiv:2210.06628*，2022年。[30] Hajime Kato、Mitsuru Nakazawa、Hsuan-Kung Yang、Mark Chen和Bj ørn Stenger。使用线性规划解析折线图图像。收录于*IEEE/CVF*冬季计算机视觉应用会议论文集，第2109–2118页，2022年。[31] Jihyung Kil、Soravit Changpinyo、Xi Chen、Hexiang Hu、Sebastian Goodman、Wei-Lun Chao和Radu Soricut。Prestu：面向场景文本理解的预训练。*arXiv*预印本 *arXiv:2209.05534*，2022年。[32] Geewook Kim、Teakgyu Hong、Moonbin Yim、JeongYeon Nam、Jinyoung Park、Jinyeong Yim、Wonseok Hwang、Sangdoo Yun、Dongyoon Han和Seunghyun Park。无OCR文档理解变换器。收录于计算机视觉–*ECCV 2022*年：第*17*届欧洲计算机视觉会议，特拉维夫，以色列，*2022*年*10*月*23*日至*27*日，论文集，第*XXVIII*部分，第498–517页。Springer，2022年。[33] Diederik P Kingma和Jimmy Ba。Adam：一种随机优化方法。*arXiv*预印本 *arXiv:1412 6980*，2014年.。[34] Hei Law和Jia Deng。CornerNet：将目标检测为成对关键点。收录于欧洲计算机视觉会议论文集，第734–750页，2018年。[35] Kenton Lee、Mandar Joshi、I ulia Turc、Hexiang Hu、Fangyu Liu、Julian Eisenschlos、Urvashi Khandelwal、Peter Shaw、Ming-Wei Chang和Kristina Toutanova。Pix2Struct：将屏幕截图解析作为视觉语言理解的预训练。*arXiv*预印本 *arXiv:2210.03347*，2022年。[36]Matan Levy、Rami Ben-Ari和Dani Lischinski。面向图表理解的分类-回归。收录于欧洲计算机视觉会议论文集，第469–484页。Springer，2022年。[37] Peizhao Li、Jiuxiang Gu、Jason Kuen、Vlad I Morariu、Handong Zhao、Rajiv Jain、Varun Manjunatha和Hongfu Liu。SelfDoc：自监督文档表征学习。收录于*IEEE/CVF*计算机视觉与模式识别会议论文集，第5652–5660页，2021年。[38] Fangyu Liu、Emanuele Bugliarello、Edoardo Maria Ponti、Siva Reddy、Nigel Collier和Desmond Elliott。跨语言与文化的视觉落地推理。收录于*2021*年自然语言处理实证方法会议论文集，第10467–10485页，2021年。[39] Fangyu Liu、Julian Martin Eisenschlos、Francesco Piccinno、Syrine Krichene、Chenxi Pang、Kenton Lee、Mandar Joshi、Wenhu Chen、Nigel Collier和Yasemin Altun。DePlot：通过图到表翻译实现单样本视觉语言推理。*arXiv*预印本 *arXiv:2212.10505*，2022年。[40] Fangyu Liu、Francesco Piccinno、Syrine Krichene、Chenxi Pang、Kenton Lee、Mandar Joshi、Yasemin Altun、Nigel Collier和Julian Martin Eisenschlos。Matcha：利用数学推理与图表反渲染增强视觉语言预训练。*arXiv*预印本 *arXiv:2212.09662*，2022年。[41] Xiaoyi Liu、Diego Klabjan和Patrick NBless。通过单个深度神经网络从图表中提取数据。*arXiv*预印本 *arXiv:1906.11906*，2019年。[42] Junyu Luo、Zekun Li、Jinpeng Wang和Chin-Yew Lin。《ChartOCR：通过深度混合框架从图表图像中进行数据提取》。载于 *IEEE/CVF*冬季计算机视觉应用会议论文集，第1917–1925页，2021年。[43] Anita Mahinpei、ZonaKostic 和 ChrisTanner。《LineCAP：面向数据可视化标题生成模型的折线图》。《style id='5'>arXiv预印本 arXiv:2207.07243，2022年。[44] AhmedMasry、Do Xuan Long、Jia Qing Tan、Shafiq Joty 和Enamul Hoque。《ChartQA：一个关于结合视觉推理和逻辑推理进行图表问答的基准测试集》。《style id='8'>arXiv预印本 arXiv:2203.10244，2022年。[45] Nitesh Methani、Pritha Ganguly、Mitesh MKhapra 和Pratyush Kumar。《PlotQA：对科学图表进行推理》。载于*IEEE/CVF*冬季计算机视觉应用会议论文集，第1527–1536页，2020年。[46] Vibhu O Mittal、Johanna D Moore、Giuseppe Carenini 和 Steven Roth。《用自然语言描述复杂图表：一个标题生成系统》。《style id='14'>《计算语言学》，24(3)：431–467，1998年。[47] Alejandro Newell、Kaiyu Yang 和 Jia Deng。《用于人体姿态估计的堆叠沙漏网络》。载于 欧洲计算机视觉会议论文集，第483–499页。Springer，2016年。[48] Jason Obeid 和 EnamulHoque。《图表转文本：通过调整Transformer模型为图表生成自然语言描述》。载于第*13*届国际自然语言生成会议论文集，第138–147页，2020年。[49] JorgePoco 和 Jeffrey Heer。《逆向工程可视化：从图表图像中恢复视觉编码》。载于《计算机图形学论坛》，第36卷，第353–363页。Wiley在线图书馆，2017年。[50] Chinmayee Rane、Seshasayee MahadevanSubramanya、Devi Sandeep Endluri、JianWu 和 C Lee Giles。《ChartReader：柱状图的自动解析》。载于信息复用与数据科学集成国际会议论文集，第318–325页。IEEE，2021年。[51] Ehud Reiter。《数据到文本系统的架构》。载于第十一届欧洲自然语言生成研讨会论文集（*ENLG07*），第97–104页，2007年。[52] Manolis Savva、Nicholas Kong、Arti Chhajta、Li Fei-Fei、Maneesh Agrawala 和 Jeffrey Heer。《Revision：图表图像的自动分类、分析与重新设计》。载于*ACM*用户界面软件与技术年度研讨会论文集，第393–402页，2011年。[53] Hrituraj Singh 和Sumit Shekhar。《STL-CQA：结合定位和编码的基于结构的Transformer，用于图表问答》。载于 自然语言处理经验方法会议论文集，第3275–3284页，2020年。[54] Arjun Srinivasan、Steven M Drucker、Alex Endert 和 JohnStasko。《通过交互式数据事实增强可视化，以促进解释和交流》。《styleid='38'>IEEE可视化与计算机图形学汇刊，25(1)：672–681，2018年。[55] Alane Suhr 和 Yoav Artzi。《NLVR2视觉偏差分析》。《style id='41'>arXiv预印本 arXiv:1909.10411，2019年。

[56] Alane Suhr、Mike Lewis、JamesYeh 和 Yoav Artzi。用于视觉推理的自然语言语料库。载于第*55*届计算语言学协会年会论文集（第*2*卷223，2017年217：短论文），页码–。[57] Chaoli Wang和 JunHan。DL4SciVis：深度学习用于科学可视化的最新综述。*IEEE*可视化与计算机图形学汇刊，2022年。[58] JianfengWang、Zhengyuan Yang、XiaoweiHu、Linjie Li、Kevin Lin、Zhe Gan、Zicheng Liu、Ce Liu和 Lijuan Wang。GIT：面向视觉与语言的生成式图像到文本转换器。*arXiv* 预印本*arXiv:2205.14100*，2022年。[59] Qianwen Wang、Zhutian Chen、Yong Wang和 Huamin Qu。ML4VIS 综述：将机器学习进展应用于数据可视化。*IEEE*可视化与计算机图形学汇刊，2021年。[60] Yun Wang、Zhida Sun、Haidong Zhang、Weiwei Cui、Ke Xu、Xiaojuan Ma 和 Dongmei Zhang。DataShot：从表格数据自动生成事实表。*IEEE*可视化与计算机图形学汇刊，26(1)：895– 905，2019年。[61] Yiheng Xu、Minghao Li、Lei Cui、Shaohan Huang、Furu Wei 和 MingZhou。LayoutLM：用于文档图像理解的文本与布局预训练。载于 第*26*届 *ACMSIGKDD* 知识发现与数据挖掘国际会议论文集，第1192–1200页，2020年。[62] Jiahui Yu、ZiruiWang、Vijay Vasudevan、Legg Yeung、MojtabaSeyedhosseini和 Yonghui Wu。CoCa：对比式图像描述生成模型是图文基础模型。*arXiv* 预印本 *arXiv:2205.01917*，2022年。[63] Jialong Zou、Guoli Wu、Taofeng Xue和Qingfeng Wu。用于图表问答的亲和度驱动的关系网络。载于 *IEEE* 多媒体与博览会国际会议论文集，第1–6页。IEEE，2020年。