TabPedia: Towards Comprehensive Visual Table Understanding with Concept Synergy

Weichao Zhao[1,2,♠,*] Hao Feng[1,*] Qi Liu[2,*] Jingqun Tang[2] Shu Wei[2] Binghong Wu[2]
Lei Liao[2] Yongjie Ye[2] Hao Liu[2,‡,†] Wengang Zhou[1,†] Houqiang Li[1] Can Huang[2]
[1] University of Science and Technology of China, [2] ByteDance Inc.,
{saruka, haof}@mail.ustc.edu.cn, {zhwg, lihq}@ustc.edu.cn
{liuqi.nero, haoliu.0128, can.huang}@bytedance.com

Abstract

Tables contain factual and quantitative data accompanied by various structures and contents that pose challenges for machine comprehension. Previous methods generally design task-specific architectures and objectives for individual tasks, resulting in modal isolation and intricate workflows. In this paper, we present a novel large vision-language model, TabPedia, equipped with a *concept synergy* mechanism. In this mechanism, all the involved diverse visual table understanding (VTU) tasks and multi-source visual embeddings are abstracted as concepts. This unified framework allows TabPedia to seamlessly integrate VTU tasks, such as table detection, table structure recognition, table querying, and table question answering, by leveraging the capabilities of large language models (LLMs). Moreover, the concept synergy mechanism enables table perception-related and comprehension-related tasks to work in harmony, as they can effectively leverage the needed clues from the corresponding source perception embeddings. Furthermore, to better evaluate the VTU task in real-world scenarios, we establish a new and comprehensive table VQA benchmark, ComTQA, featuring approximately 9,000 QA pairs. Extensive quantitative and qualitative experiments on both table perception and comprehension tasks, conducted across various public benchmarks, validate the effectiveness of our TabPedia. The superior performance further confirms the feasibility of using LLMs for understanding visual tables when all concepts work in synergy. The benchmark ComTQA has been open-sourced at https://huggingface.co/datasets/ByteDance/ComTQA. The source code and model also have been released at https://github.com/zhaowc-ustc/TabPedia.

1 Introduction

With the rapid advancement of digital technology, numerous paper documents must be converted into electronic formats for efficient storage and utilization. Tables, as indispensable components of documents, play a vital role in summarizing facts and quantitative data [1, 2]. The compact yet informative nature of tables makes them advantageous for various applications, thereby attracting widespread research attention toward Visual Table Understanding (VTU). VTU generally encompasses four subtasks: *Table Detection* (TD), which locates tables within document images; *Table Structure Recognition* (TSR), which parses the structure of tables in table-centric images; *Table Querying* (TQ), which recognizes the structure of a table from an entire image at a given location, a task that remains underexplored in the previous works; and *Table Question Answering* (TQA), which

[*] Equal contribution. ♠ Interns at ByteDance. ‡ Project lead.

[†] ✉ Corresponding authors: Wengang Zhou and Hao Liu.

Preprint. Under review.

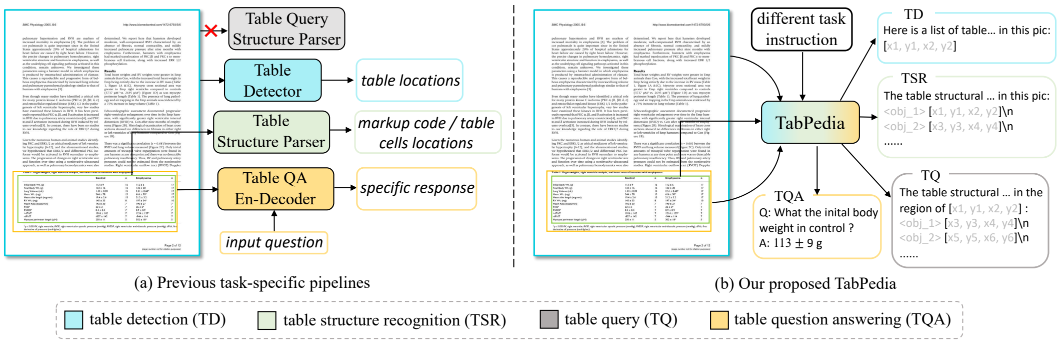


Table Query Structure Parser
Table Detector
table locations
Table Structure Parser
markup code / table cells locations
Table QA En-Decoder
specific response
input question
different task instruction
TabPedia
TD
Here is a list of table... in this pic:
[x1, y1, x2, y2]
TSR
The table structural ... in this pic:
<obj_1> [x1, y1, x2, y2]\n
<obj_2> [x3, y3, x4, y4]\n
......
TQA
Q: What the inital body weight in control ?
A: 113 ± 9 g
TQ
The table structural ... in the region of [x1, y1, x2, y2] :
<obj_1> [x3, y3, x4, y4]\n
<obj_2> [x5, y5, x6, y6]\n
......
table detection (TD)
table structure recognition (TSR)
table query (TQ)
table question answering (TQA)

(a) Previous task-specific pipelines

(b) Our proposed TabPedia

Figure 1: Comparison with previous task-specific pipelines for visual table understanding. In contrast to design different architectures for various table tasks, our TabPedia effectively performs these tasks in a unified framework through delicately leveraging the understanding capability of LLMs.

answers questions based on table contents. These tasks pose challenges from various perspectives due to the need for representations at different visual-semantic granularities and hierarchies.

Given the success achieved, many pioneering works have mainly centered on the specific subtask with various task-specific architectures, as shown in Fig. 1 (a). For visual table perception tasks such as TD and TSR, one of most adopted approaches is in the detection manner [3–9]. In contrast, generative vision-language models [10–13] are often employed to generate answers conditioned on the semantic content of tables for TQA task. Specifically, Vision Transformers (ViT) [14] pretrained on CLIP [15] or EVA-CLIP [16], Swin-Transformer [17], and similar models serve as vision encoders, while language models operate in either encoder-decoder [18, 19] or decoder-only frameworks [11, 20–22]. Besides, recent fast-growing Large Vision Language Models (LVLMs) [11, 13, 23–34] have shown their powerful capabilities to perceive and understand visual clues by integrating instruction following of Large Language Models (LLMs) [35–39]. Despite impressive progress, the *status quo* begs for a question: "*Can we leverage the advantages of LVLMs to solve all the VTU tasks once and for all?*"

A straightforward solution would be to train the LVLM directly using all the VTU data. However, aside from the diverse table structure and the various relations of table contents, it remains a nontrivial issue due to two cruxes of table parsing and understanding: (i) discrepancy between the representation formats (two-dimensional structure VS. one-dimensional sequence); (ii) required image resolutions. Although some works [40–42] represent table structure in markup formats like HTML, XML, Markdown, or LATEX. However, they neglect spatial coordinates for cells and only encode logical relationships implicitly. The generated code contains extensive formatted information from different markup languages, increasing output length and potentially causing parsing issues with illegal grammars.

To attack above issues, we in this paper propose a novel LVLM tailored for comprehensive VTU, TabPedia, to effectively solve all VTU tasks in a unified framework, as shown in Fig. 1 (b). More concretely, we employ dual vision encoders, namely ViT-L [15] and Swin-B [43], to encode the global and fine-grained local information in the low- and high-resolution formats of the input image respectively, acquiring multi-source visual embeddings. Here, all the involved VTU tasks and multi-source visual embeddings are abstracted as *concepts* and *concept synergy* mechanism is implemented by introducing the *mediative tokens* to the LLM in our model. Thanks to this mechanism, all the concepts in TabPedia can work in synergy flexibly. Quantitative and qualitative experimental results on both table perception and comprehension tasks across various public benchmarks confirm the effectiveness of our proposed TabPedia. To further investigate the potential of our model in more challenging and realistic scenarios, we establish a new and comprehensive table VQA benchmark, ComTQA, featuring round 1,500 images and 9,000 QA pairs.

Our contributions are summarized as follows,

- We propose a novel large vision-language model, TabPedia, to integrate various VTU tasks into a unified framework, including TD, TSR, TQ and TQA. Specifically, TabPedia fully leverages the comprehensive capabilities of LLMs to fertilize complex table understanding.
- We design a concept synergy mechanism to harmonize both table perception and comprehension tasks. Through introducing the meditative tokens into our framework, TabPedia

adaptively enables useful information in multi-source visual embeddings and task instructions, generating accurate and plausible responses.

- Extensive quantitative and qualitative experiments validate the effectiveness of our proposed TabPedia across various tasks and benchmarks. To further exploit the potential of our model in more complex scenarios, we build a new table VQA benchmark, ComTQA, involving multiple answers, mathematical calculation and logical reasoning, *etc*.

2 Related Work

2.1 Table Recognition

Table recognition is generally divided into table detection, table structure recognition and table content recognition In our work, table content recognition is beyond our scope.

For TD task, the earliest approaches are rule-based methods for locating tables inside documents [44–46]. With the rapid advances in deep learning, numerous CNN-based methods show impressive performance. Most of these methods directly adopt top-down object detection frameworks to solve this problem [5, 47–52]. For instance, Sun *et al.* [52] adopt Faster R-CNN [52] to detect table boxes and the corresponding corner boxes simultaneously, and then adjust table boundaries according to the detected corners. Some other methods model each document image as a graph and formulate TD as a graph labeling problem [53–55]. In addition, TATR [9] first applies the transformer-based detector, DETR [56], to improve the detection accuracy without special customization.

For TSR task, one of the most common modeling approaches is still to regard it as some form of object detection [3–5, 9, 57–59]. Among them, DeepDeSRT [4] and TableNet [60] are both representative works exploring semantic segmentation to obtain table cell boundaries. TATR [9] first proposes to utilize DETR for this task. TSRFormer [58] introduces a cross-attention module into the DETR framework to improve the localization accuracy of row/column separators. Some other methods attempt to parse table structure via modeling relationship among different table elements [61–63]. As the most relevant to our approach, markup generation-based methods directly generate markup (HTML or LaTeX) sequences from raw table images [41, 64]. EDD [64] introduces a cell decoder and a structures decoder to generate HTML codes. OmniParser [41] further integrates three task-specific decoders to enhance the table structure representation.

While the previous methods have achieved promising results on table perceptive tasks, they are still limited in table intricate content understanding. In our work, we jointly exploit table perception and comprehension tasks in a unified framework, concurrently enriching visual table understanding.

2.2 Large Vision-Language Models

LVLMs aim to equip LLMs [29, 36, 38, 39, 65] with visual comprehension capability. The mainstream approaches attempt to connect visual encoders and LLMs with intermediate modules such as simple Projectors [30], QFormer [25], Perceiver Resamplers [23], achieving visual language understanding through pre-training alignment and instruction fine-tuning. For text-rich document scene, several works [10, 13, 40, 41, 66–68] propose to enhance the LVLMs' capabilities in understanding textual elements (text-centric VQA, OCR, text spotting, *etc.*). Among them, TextMonkey [12] employs shifted window attention and token resampler module to improve the training process. DocOwl-1.5 [40] collects a comprehensive dataset DocStruct4M to support unified structure learning.

Despite achieving extraordinary progress on visual understanding, existing LVLMs still face challenges in two-dimensional table parsing and understanding. In this paper, we propose a unified framework to concurrently achieve table perception and comprehension with the support of LLMs.

2.3 Additional Tokens

In the trend of Transformer-based approaches, extending the input sequence with special tokens is popularized for various intentions, such as extracting task-specific information [14, 56], providing extra information [69, 70] or improving model performance [71–74]. For instance, ViT [14] utilizes [CLS] token for classification. Similarly, DETR [56] proposes object queries for detection. ATR [70] adopts tape tokens to obtain useful information from a memory bank. In addition, the Memory Transformer [71] presents a simple approach to improve translation performance by attaching trainable memory tokens after the token sequence. Darcet *et al.,* [73] further attempt to add extra tokens in ViT-based frameworks, *e.g.,* CLIP [15] and DINOv2 [75], thus improving visual tasks. In our work, we

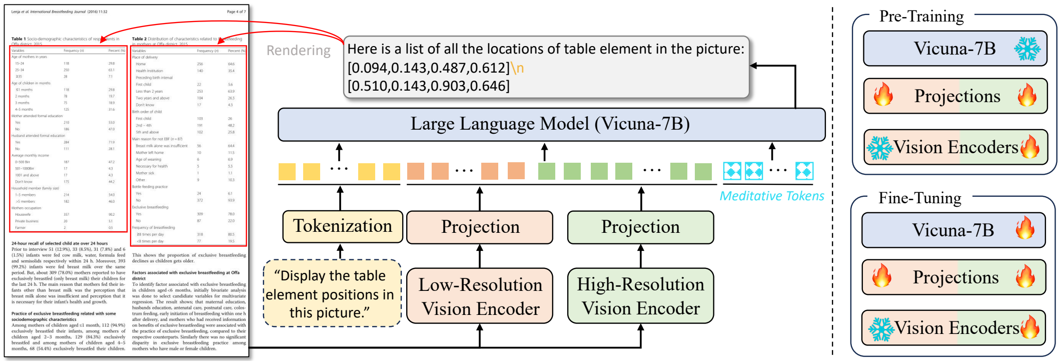

Rendering
Here is a list of all the locations of table element in the picture:
[0.094,0.143,0.487,0.612]\n
[0.510,0.143,0.903,0.646]
Large Language Model (Vicuna-7B)
Meditative Tokens
Tokenization
Projection
Projection
"Display the table element positions in this picture."
Low-Resolution Vision Encoder
High-Resolution Vision Encoder
Pre-Training
Vicuna-7B
Projections
Vision Encoders
Fine-Tuning
Vicuna-7B
Projections
Vision Encoders

Figure 2: The illustration of our proposed TabPedia. Given the input image, TabPedia feeds it into both vision encoders attached projections to extract different granular features. Then, the visual tokens are combined with instruction-derived tokens, and fed into the LLM. The LLM leverages its powerful understanding ability to generate a plausible response.

inherit this spirit and design meditative tokens to enhance TabPedia's perceptive and comprehensive capability for visual tables.

3 Method

As shown in Fig 2, we present an overview of TabPedia. The overall training pipeline consists of two phases. Concretely, the pre-training stage aims to align the visual features to the large language model, and the fine-tuning stage focuses on visual table-aware understanding. In the following, we elaborate on the architecture of TabPedia, followed by the exposition of its two training phases.

3.1 Model Architecture

High-Resolution Vision Encoder. As proved by previous methods [43, 76, 77], the high-resolution image is critical to ensuring that the LLMs could grasp rich visual information. Following Donut [43], we adopt Swin-B [17] to encode the high-resolution format of input image. Given the input RGB image I, we first resize it to pre-defined high-resolution scale of $\mathrm{H} \times \mathrm{W}$, denoted as I_h. By default, both H and W are set to 2,560 and 1,920, respectively. Notably, we maintain the aspect ratio during the resizing process to prevent distortion of table contents and structures. Then, the resized image I_h is fed into the vanilla Swin Transformer initialized from [43] to obtain a feature map V_h downsampled by a factor of 1/32, each token with 1,024 dimension.

Low-Resolution Vision Encoder. To keep the overall layout information, the raw image is also resized to a low-resolution one denoted as I_l. We choose the pre-trained CLIP visual encoder ViT-L/14 [15] to encode the low-resolution image with 224×224, which has been pre-trained on 400 million image-text pairs sourced from the open-world data, thereby embedding extensive world knowledge into its pretrained weights. To preserve its generalization ability, we keep it frozen during the whole training procedure. The output sequence V_l is composed of 256 tokens, each with 1024 dimension.

Projections. The projections are designed to align visual tokens with the input token dimension of the subsequent large language model [65]. For the high-resolution feature map V_h, due to the limitation of input text length, we employ a 2D convolutional layer with a kernel size of 3 and a stride of 2, and then flatten it into $\frac{\mathrm{H}}{64} \times \frac{\mathrm{W}}{64}$ tokens, denoted as $\hat{V}_h$. For the low-resolution visual features V_l, inspired from the paradigm of advanced LVLMs [29, 30], we adopt a linear layer to project visual tokens, denoted as $\hat{V}_l$.

Concept Synergy. Given the massive visual tokens and the embedding of textual instruction Q, we utilize Vicuna-7B [65] as LLM to generate its response. Taking into account the discrepancy of table perception and comprehension tasks, we introduce *meditative tokens* M to implement the concept synergy for the LLM, which adaptively enable different region of visual tokens and understand the intentions of specific task question. Finally, we construct the whole input sequence as $X =$

Table 1: Summary of training data statistics in the fine-tuning stage.

Dataset	Subset	Task	Num
PubTab1M	PubTab1M-Det	TD	460k
	PubTab1M-Str	TSR,TQA	759k
	PubTab1M-Syn	TQ	381k
FinTabNet	–	TSR,TQA	78k
PubTabNet	–	TSR	434k
WTQ	–	TQA	1k
TabFact	–	TQA	9k

Table 2: Different task types and their instruction examples.

Task	Example
TD	"Give me the areas where table element's locations in this picture."
TSR	"Parse the structural information of the cropped table in this picture."
TQ	"Parse the table structure within the region [0.095, 0.673, 0.869, 0.851] in this picture."
TQA	"What was the lowest stock price in the fourth quarter of 2010?"

[Q, <IMG_S> ; $\hat{V}_l$; <IMG_SEP> ; $\hat{V}_h$; <IMG_E> ; M], where [;] means the concatenation operation. <IMG_S>, <IMG_E> and <IMG_SEP> are learnable special tokens, that denote the start and end of visual tokens as well as the separation of different resolution tokens, respectively.

Objective. Since TabPedia is trained to predict the next tokens like other LLMs, it is optimized by maximizing the likelihood of prediction loss at training time.

3.2 Pre-training

To enable the capable of vision encoders to capture text-rich information from high-resolution images and aligning embedding space with the large language model [65], we first perform extensive text-aware pre-training. As shown in Fig. 2, we jointly optimize the high-resolution visual encoder with both projectors, while freezing the large language model and low-resolution vision encoder. Specifically, followed by [10], our pre-training procedure involves a variety of perception tasks, *i.e.*, text detection [78], recognition [79], spotting [80], long-text reading [43] and image captioning [81]. The first four tasks focuses on the various document images, while the last one targets natural scene images. These comprehensive tasks endow the vision encoders of TabPedia to effectively perceive textual and visual information from both document and natural scene images. More detailed pre-training settings about dataset and experiment could be referred to [10].

3.3 Table-aware Fine-tuning

Through pre-training, TabPedia could well understand text and structure of diverse document images but cannot follow instructions to perform different table understanding tasks. In order to enhance the model capability of instruction following, we *first* construct a large-scale dataset for visual table understanding. We will elaborate on the dataset construction in the Sec. 4. Based on this dataset, we introduce four table-related tasks, *i.e.*, TD [9], TSR [5, 9, 64], TQ and TQA [5, 9, 82, 83] to simultaneously cultivate the perception and comprehension capabilities. In this stage, we further unfreeze the LLM and fine-tune the entire framework except the low-resolution vision encoder.

4 Dataset Construction

In this section, we aim to introduce the collected instruction following dataset. The entire data is derived from five public datasets, including PubTab1M [9], FinTabNet [5], PubTabNet [64], WikiTableQuestions (WTQ) [82] and TabFact [83]. Among them, PubTab1M [9] contains two subsets, *i.e.*, PubTab1M-Detection (PubTab1M-Det) and PubTab1M-Structure (PubTab1M-Str). Moreover, since the table images in PubTab1M-Str are cropped from PubTab1M-Det, we transform the annotations of the table structure in PubTab1M-Str into the original images and synthesize a new subset PubTab1M-Syn, which could be utilized for TQ task. The statistical data are summarized in Tab. 1. To ensure the instruction diversity, we generate multiple instructions for each task using GPT3.5 [21]. In Tab. 2, we display one exemplar about user's question for each table task. We will provide a detailed exposition of them in the following.

Table Detection (TD). As a fundamental task, TD task targets to detect all table locations in a document image. Previous methods [3, 6, 9] mainly utilize DETR [56] or variants of R-CNN [84–86] to predict numerous overlapping bboxes, that inevitably needs complex post-processing, such as non-maximization suppression (NMS), to generate final results. In contrast, we employ LLM to directly generate the locations of instance tables in the format of "[x1, y1, x2, y2]", where x1, y1, x2, y2 represent the normalized coordinates of the top-left and bottom-right of the corresponding bbox. Moreover, to facilitate detection results for multiple tables, we split multiple table positions with the special symbol "\n" in the output response. We adopt PubTab1M-Det [9] to perform TD task, where images are collected from PDF documents with different scale and rotation types of tables.

Table Structure Recognition (TSR). The TSR targets to parse table structure in terms of rows, columns and cells. HTML and Markdown codes are mainly two kinds of text sequences used to represent a table. HTML could represent all kinds of tables, with or without cells spanning multiple rows and grids, but they contain massive markup grammars *i.e.*, "`<div></div>`" and "`<td></td>`", resulting in excessively lengthy output responses. Compared with HTML, Markdown represents a table more succinctly, but it cannot represent cells spanning multiple rows or columns. By weighing the simplicity of the output and the completeness of the table parsing, we propose a canonical table structure representation based on the detection format. Inspired by [9], we jointly adopt five object classes to model TSR, including *table column*, *table row*, *table column header*, *table projected row header* and *table spanning cell*. To better understanding, we display a representative sample in Appendix B. Taking into account the serialized output of the LLM, we represent the table structure with a series of "[object] [x1, y1, x2, y2]", which are also separated by "\n". Notably, we standardize the order of the output objects to ensure uniqueness of the table parsing results.

We select the PubTab1M-Str [9], FinTabNet [5] and PubTabNet [64] to support the TSR task, where tables are collected from scientific and financial articles. These datasets contain pairs of table images and HTML annotations. We convert HTML codes into our designed annotation format using the pre-processing tool offered by [9].

Table Querying (TQ). Different from recognizing table structure from the cropped table-centric images in TSR task, the TQ task directly parses the table from the original document image based on the given table location. This task is more challenging due to the degradation of the table's resolution and the interference of other document contents around it. Moreover, this task could potentially be combined with TD task to enable automatic parsing of all table structure information in original images. Therefore, we introduce this task to fully unlock the comprehension capabilities of large language models for visual table understanding. For the annotation of table parsing, we adopt the same format as TSR. Since there is no readily available dataset, we synthesize a large amount of available data based on the annotations from PubTab1M [9], namely PubTab1M-Syn.

Table Question Answering (TQA). TQA aims to provide precise answers through table understanding and reasoning. For both public TQA datasets, *i.e.*, WTQ [82] and TabFact [83], the table images are collected from wikipedia tables with pairs of content-related question and answer. Thus, we could directly apply these available data to support this task. However, the images of current TQA data are rendered from text-based tables with variations in background color and font size, resulting in poor generalization in real-world tables. In addition, the TQA data volume lags far behind other tasks. To alleviate these obstacles, we generate numerous TQA data with partial images in FinTabNet [5] and PubTab1M [9] by employing the powerful multi-modal understanding capabilities of Gemini Pro [87]. We provide more detailed descriptions of the procedure in the Appendix A.1

To better evaluate TQA performance of various models on real-world table images, we build a complex TQA dataset (ComTQA) based on test set of FinTabNet [5] and PubTab1M [9]. Compared to WTQ and TabFact, ComTQA has more challenging questions, such as multiple answers, mathematical calculations, and logical reasoning. In total, we annotate $\sim$9k high-quality QA pairs from $\sim$1.5k images by expert annotation. More statistics about ComTQA could be found in the Appendix A.2.

5 Experiment

5.1 Implementation Details

Parameter Settings. For the hyper-parameters in model design, the number of meditative tokens is set to 256. The max length of text sequence is set to 4000 to satisfy task requirements. To implement TabPedia, we adopt a cosine schedule with one-cycle learning rate strategy [88]. In the pre-training phase, the learning rate warms up in the first 2% of the training process and then decreases from the peak rate (1e-3) with batch sizes of 64. In the fine-tuning phase, we set the peak learning rate as 5e-6 with batch sizes of 16. We employ the AdamW optimizer [89] in both phases. All experiments are implemented by PyTorch [90] and trained on 16× A100 GPUs.

Table 3: Comparison with the existing best table detection model TATR [9]. NMS denotes Non-Maximum Suppression.

Method	Backbone	NMS	IoU@0.75		
			Precision	Recall	F1
TATR [9]	Faster R-CNN	✓	92.7	86.6	89.5
	DETR	✓	**98.8**	98.1	**98.4**
TabPedia	LVLM	✗	98.5	**98.4**	**98.4**

Table 5: Quantitative results on two subsets of PubTab1M [9], including PubTab1M-Str and PubTab1M-Syn.

(a) Comparison with the task-specific model, TATR [9] on TSR task. "Cropped" denotes utilizing cropped table-centric images.

Method	Backbone	Image	NMS	PubTab1M-Str			
				$\text{GriTS}_{\text{Top}}$	$\text{GriTS}_{\text{Cont}}$	$\text{GriTS}_{\text{Loc}}$	S-TEDS
TATR [9]	Faster R-CNN	Cropped	✓	86.16	85.38	72.11	–
	DETR	Cropped	✓	**98.46**	**97.81**	**97.81**	**97.65**
TabPedia (TSR)	LVLM	Cropped	✗	96.52	96.73	95.54	95.66

(b) Quantitative results on both TQ and TD+TQ tasks.

Method	Image	NMS	Task	PubTab1M-Syn			
				$\text{GriTS}_{\text{Top}}$	$\text{GriTS}_{\text{Cont}}$	$\text{GriTS}_{\text{Loc}}$	S-TEDS
TabPedia	**Raw**	✗	TQ	96.04	96.23	94.95	95.07
			TD+TQ	94.54	94.63	93.25	93.38

Table 4: Comparison with end-to-end TSR methods on two datasets. "$*$" represents the results reported by [41].

Method	Input Size	PubTabNet	FinTabNet
		S-TEDS	S-TEDS
Donut [43]*	1,280	25.28	30.66
EDD [64]	512	89.90	90.60
OmniParser [41]	1,024	90.45	91.55
TabPedia	2,560	**95.41**	**95.11**

Table 6: Comparison with existing LVLMs on TQA task. "$*$" denotes the results obtained through the open-source checkpoint or API of the closed-source model. ComTQA is our released new benchmark. The second best methods are underlined.

Method	Input Size	WTQ	TabFact	ComTQA
		Acc	Acc	Acc
TextMonkey [12]	896	37.9	53.6	13.9*
Monkey [93]	896	25.3*	49.8	–
Cogagent [94]	1,120	30.2*	51.7*	–
DocOwl 1.5 [40]	1,344	39.8	**80.4**	18.5*
GPT4V [95]	645	45.5*	69.3*	27.2*
Gemini Pro [87]	659	32.3*	67.9*	29.3*
Xcomposer2 [96]	511	28.7	62.3	–
TabPedia	2,560	**47.8**	71.3	**53.5**

Datasets. In order to comprehensively evaluate the capability of TabPedia, we employ multiple benchmarks for each task. For performance assessment, we set the temperature parameter as 0.2 in both quantitative and qualitative evaluations. For TD task, PubTab1M-Det [9] contains 57,125 images for testing. For TSR task, FinTabNet [5], PubTabNet [64] and PubTab1M-Str [9] are adopted for evaluation with 9,289, 9,115 and 93,834 testing samples, respectively. For TQ task, the synthetic dataset PubTab1M-Syn [9] also provides 47,186 samples for testing. For TQA task, WTQ [82], TabFact [83] and our annotated ComTQA contain 4,343, 12,722 and 9,070 QA pairs, respectively.

Evaluation Metrics. For TD task, we report the results with object detection metrics, including precision, recall and f1-score with IoU@0.75. For both TSR and TQ tasks, we utilize Structure Tree-EditDistance-based Similarity (S-TEDS) [64], which evaluates table similarity of structural aspects in HTML format. The metric represents the HTML table as a tree, and the TEDS score is computed through the tree-edit distance between the ground truth and predicted trees. In order to convert the results of TabPedia into HTML format, we employ the post-processing algorithm provided by [9]. Moreover, we report the recently proposed GriTS metrics [91] for PubTab1M-Str to align its original metric. Different from S-TEDS, GriTS represents tables as matrices, better capturing the two-dimensional structure and the orders of cells in a table. Further, GriTS enables TSR to be assessed from multiple perspectives, with $\text{GriTS}_{\text{Top}}$ measuring cell topology recognition, $\text{GriTS}_{\text{Cont}}$ measuring cell content recognition, and $\text{GriTS}_{\text{Loc}}$ measuring cell location recognition. For TQA task, we adopt the accuracy metric where the response generated by the model is judged correct if it contains the string present in the ground truth [92].

5.2 Quantitative Results

We conduct quantitative evaluations of current state-of-the-art methods for specific tasks in perception and comprehension, comparing them to our proposed TabPedia.

Evaluation on TD. In Tab. 3, we compare TabPedia with the previous state-of-the-art method, TATR [9]. TATR performs the table detection with two classic visual detection backbones, *i.e,* DETR [56] and Faster R-CNN [85]. Compared with them, TabPedia outperforms Faster R-CNN with a notable margin and achieves competitive performance with DETR. Notably, since TabPedia directly generates the independent locations of instance tables without densely overlapped bboxes, there are no extra post-processing operations involved, *i.e.*, Non-Maximum Suppression (NMS). This advantage could enable TabPedia to perform more complex table understanding, such as parsing all tables by combining TD and TQ tasks.

Evaluation on TSR. Tab. 4 reports the performance of TSR task compared to end-to-end TSR models on PubTabNet and FinTabNet datasets. Specifically, the OCR-free model Donut [43] is fine-tuned for TSR with the official default training configuration. Although OmniParser [41] integrates multiple visually-situated text parsing tasks into a unified framework, it adopts three isolated decoders to perform different tasks. Compared with OmniParser, TabPedia consistently surpasses it with 4.96% and 3.56% S-TEDS on both datasets, respectively. In Tab. 5a, TATR as the task-specific method, shows high performance with the DETR architecture. Our proposed TabPedia, a generic model for tasks involving both perception and comprehension, still achieves comparable performance without the need for complex post-processing. These results highlight the exceptional capability of TabPedia.

Evaluation on TQ. As a new and unexplored task, the TQ task aims to parse table structures with the specific location directly from the raw image without additional cropping. In the first row of Tab. 5b, we provide a strong baseline with 96.04% and 95.07% on $\mathrm{GriTS}_{\mathrm{Top}}$ and S-TEDS, respectively, which nearly reaches the same performance as parsing from the cropped images under the interference of the document content around the table. Furthermore, we integrate both TD and TQ tasks in the form of multi-round dialogue, which endows TabPedia to directly parse all existing tables in a document image. We report the final result in the second row of Tab. 5b. These impressive results demonstrate that TabPedia has the potential to enable more holistic table understanding.

Evaluation on TQA. Due to the complex structure of tables and the dense text, the understanding of the table contents remains a challenging issue. To thoroughly evaluate the performance of the understanding of table content and structure, we adopt two public benchmarks, *i.e.*, WTQ [82] and TabFact [83], and our collected dataset ComTQA, as shown in Tab. 6. On the WTQ and TabFact, TabPedia achieves promising performance among the open and close sources LVLMs. In contrast to existing benchmarks, ComTQA contains real-world table images with more complex questions. It is observed that current LVLMs show poor performance due to the incomplete understanding of real-world table structures. Compared with them, TabPedia achieves the optimal result with a notable margin, which demonstrates the effectiveness of jointly learning perception and comprehension tasks.

5.3 Qualitative Results

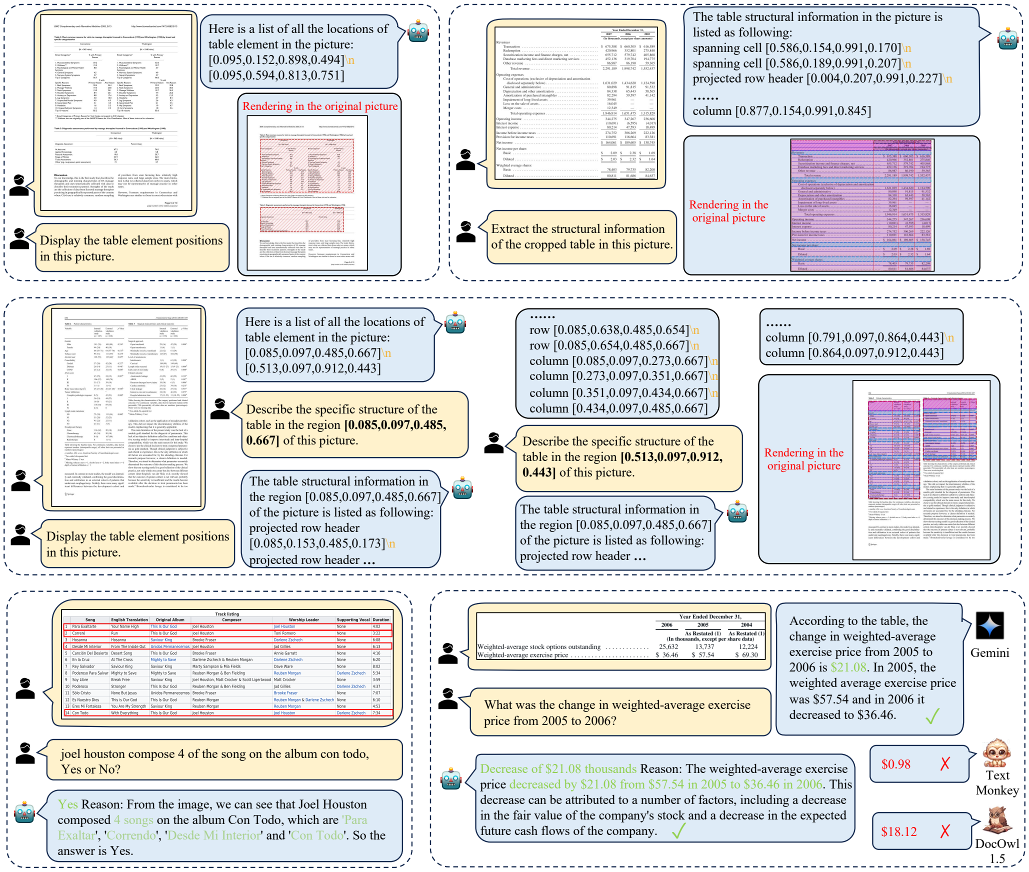
We further conduct qualitative evaluation on TabPedia's perception and comprehension capabilities. Firstly, we show the perception capability of TabPedia with solely TD and TSR tasks, as illustrated in the first row of Fig. 3. TabPedia accurately generates reliable and formatted results, which are rendered to the original image for better observation. Secondly, TabPedia performs a complex task to directly parsing all table structure information in a document image by integrating instructions of TD and TQ tasks within a multi-round dialogue. As shown in the second row of Fig. 3, the example indicates that TabPedia is capable of exploring more holistic visual table understanding. In the last row, we display the table comprehensive capability of TabPedia. It is observed that the response not only contains concise and reliable answer, but also provides the specific contents in the table to support its answer. Especially, TabPedia even acquires certain math calculation ability to capture the connections among table contents, as shown in the bottom right example in Fig. 3. These results demonstrate Tabpedia's powerful multimodal comprehension capabilities. We also display more visualization results in the Appendix D.

5.4 Ablation Studies

In this section, we conduct ablation studies to validate the effectiveness of core settings and components in TabPedia. All experiments are conducted on three datasets across three tasks: PubTab1M-Det [9], FinTabNet [5] and WTQ [82].

Necessity of Meditative Tokens. In Tab. 8, we conduct the experiment to investigate the impact of adding meditative tokens in TabPedia. It is observed that adding meditative tokens significantly improves TabPedia's capabilities of table perception and comprehension.

What Information Matters for Meditative Tokens? We sample 100 test cases for each task and report the averaged numeric importance of high- and low-resolution vision tokens when they are attended by the meditative tokens for different tasks in the Tab. 9. Specifically, for the various VTU



Here is a list of all the locations of table element in the picture: [0.095,0.152,0.898,0.494]\n [0.095,0.594,0.813,0.751]

Display the table element positions in this picture.

The table structural information in the picture is listed as following:
spanning cell [0.586,0.154,0.991,0.170]\n
spanning cell [0.586,0.189,0.991,0.207]\n
projected row header [0.004,0.207,0.991,0.227]\n
……
column [0.877,0.154,0.991,0.845]

Extract the structural information of the cropped table in this picture.

Here is a list of all the locations of table element in the picture: [0.085,0.097,0.485,0.667]\n [0.513,0.097,0.912,0.443]

Describe the specific structure of the table in the region **[0.085,0.097,0.485, 0.667]** of this picture.

The table structural information in the region [0.085,0.097,0.485,0.667] of the picture is listed as following: projected row header [0.085,0.153,0.485,0.173]\n projected row header …

Display the table element positions in this picture.

……
row [0.085,0.638,0.485,0.654]\n
row [0.085,0.654,0.485,0.667]\n
column [0.085,0.097,0.273,0.667]\n
column [0.273,0.097,0.351,0.667]\n
column [0.351,0.097,0.434,0.667]\n
column [0.434,0.097,0.485,0.667]

Describe the specific structure of the table in the region **[0.513,0.097,0.912, 0.443]** of this picture.

The table structural information in the region [0.085,0.097,0.485,0.667] of the picture is listed as following: projected row header …

……
column [0.791,0.097,0.864,0.443]\n
column [0.864,0.097,0.912,0.443]

Track listing

	Song	English Translation	Original Album	Composer	Worship Leader	Supporting Vocal	Duration
1	Para Exaltarte	Your Name High	This Is Our God	Joel Houston	Joel Houston	None	4:02
2	Correré	Run	This Is Our God	Joel Houston	Toni Romero	None	3:22
3	Hosanna	Hosanna	Saviour King	Brooke Fraser	Darlene Zschech	None	6:08
4	Desde Mi Interior	From The Inside Out	Unidos Permanecemos	Joel Houston	Jad Gillies	None	6:13
5	Canción Del Desierto	Desert Song	This Is Our God	Brooke Fraser	Annie Garratt	None	4:16
6	En la Cruz	At The Cross	Mighty to Save	Darlene Zschech & Reuben Morgan	Darlene Zschech	None	6:20
7	Rey Salvador	Saviour King	Saviour King	Marty Sampson & Mia Fields	Dave Ware	None	8:02
8	Poderoso Para Salvar	Mighty to Save	Mighty to Save	Reuben Morgan & Ben Fielding	Reuben Morgan	Darlene Zschech	5:34
9	Soy Libre	Break Free	Saviour King	Joel Houston, Matt Crocker & Scott Ligertwood	Matt Crocker	None	3:59
10	Poderoso	Stronger	This Is Our God	Reuben Morgan & Ben Fielding	Jad Gillies	Darlene Zschech	4:37
11	Sólo Cristo	None But Jesus	Unidos Permanecemos	Brooke Fraser	Brooke Fraser	None	7:07
12	Es Nuestro Dios	This is Our God	This Is Our God	Reuben Morgan	Reuben Morgan & Darlene Zschech	None	6:10
13	Eres Mi Fortaleza	You Are My Strength	Saviour King	Reuben Morgan	Reuben Morgan	None	4:53
14	Con Todo	With Everything	This Is Our God	Joel Houston	Joel Houston	Darlene Zschech	7:34

joel houston compose 4 of the song on the album con todo, Yes or No?

Yes Reason: From the image, we can see that Joel Houston composed 4 songs on the album Con Todo, which are 'Para Exaltar', 'Correndo', 'Desde Mi Interior' and 'Con Todo'. So the answer is Yes.

	Year Ended December 31,		
	2006	2005	2004
		As Restated (1)	As Restated (1)
	(In thousands, except per share data)		
Weighted-average stock options outstanding	25,632	13,737	12,224
Weighted-average exercise price	$ 36.46	$ 57.54	$ 69.30

What was the change in weighted-average exercise price from 2005 to 2006?

According to the table, the change in weighted-average exercise price from 2005 to 2006 is $21.08. In 2005, the weighted average exercise price was $57.54 and in 2006 it decreased to $36.46. ✓

Decrease of $21.08 thousands Reason: The weighted-average exercise price decreased by $21.08 from $57.54 in 2005 to $36.46 in 2006. This decrease can be attributed to a number of factors, including a decrease in the fair value of the company's stock and a decrease in the expected future cash flows of the company. ✓

$0.98 ✗

$18.12 ✗

Figure 3: Qualitative results of TabPedia on diverse tasks. The first row shows its perception capability on both TD and TSR tasks. The second row further exhibits TabPedia's powerful ability by employing multiple instructions of different tasks. The bottom row showcases TabPedia's accurate responses based on intricate contents in visual tables. Zoom in for best view.

tasks, we calculate the averaged attention scores (across all layers and attention heads) from the LLM decoder, which indicates the extent to which the meditative tokens focus on either high- or low-resolution visual tokens. For the TSR and TQ tasks, the meditative tokens pay significantly more attention to the high-resolution visual encoder tokens. We attribute this to the fact that both tasks require more fine-grained visual information to be "deliberated" in order to construct the dense table structure. In contrast, for the TD and TQA tasks, the two visual encoders contribute almost equally to the information attended to by the meditative tokens, validating the importance of both vision encoders for these tasks.

Contributions of Different Tokens. In the Tab. 7, we calculate the averaged scores of the TabPedia-generated answers with respect to meditative tokens, high-resolution visual tokens, and low-resolution visual tokens across all the attention maps from the LLM, respectively. One can observe that the meditative tokens contribute the most information to the generation of satisfactory answers, which demonstrates that the proposed meditative tokens are indispensable and effective. We also provide a detailed analysis of the attention map of meditative tokens in Fig. D4 of Appendix. D.

Table 7: Contributions of different tokens.

Task	Meditative tokens	High-res visual tokens	Low-res visual tokens
TD	0.65	0.16	0.19
TSR	0.64	0.12	0.24
TQ	0.71	0.11	0.19
TQA	0.56	0.18	0.25

Impact of Dual Vision Encoders. As shown in Table 11, we explore the impact of different vision encoders that capture global and local information from input images at various resolutions. The high-resolution encoder extracts intricate details from text-rich images, outperforming the low-resolution encoder, which struggles with nuanced visual representations in complex document images. Different

Table 8: Impact of meditative tokens in TabPedia.

meditative token	**PubTab1M-Det**	**FinTabNet**	**WTQ**
	Precision	**S-TEDS**	**Acc**
×	93.5	92.17	43.2
✓	**98.5**	**95.11**	**47.8**

Table 9: Impact of different training strategies on the low-resolution vision encoder.

Task	**High-res visual tokens**	**Low-res visual tokens**
TD	0.49	0.51
TSR	0.71	0.29
TQ	0.73	0.27
TQA	0.51	0.49

Table 10: Impact of different training strategies on low-resolution vision encoder.

Low-Res Encoder	**PubTab1M-Det**	**FinTabNet**	**WTQ**
	Precision	**S-TEDS**	**Acc**
frozen	**98.5**	95.11	**47.8**
unfrozen	98.4	**95.11**	46.4

Table 11: Impact of dual vision encoders.

High-Res Encoder	Low-Res Encoder	**PubTab1M-Det**	**FinTabNet**	**WTQ**
		Precision	**S-TEDS**	**Acc**
✓		96.5	93.6	44.9
	✓	86.2	81.3	24.7
✓	✓	**98.5**	**95.11**	**47.8**

tasks may require distinct visual cues, so dual vision encoders offer flexibility. For instance, TQA tasks need detailed table information, while TSR tasks depend on global layout. The low-resolution encoder provides comprehensive layout insights, complementing the high-resolution encoder's limited receptive field. Our results demonstrate that combining both encoders enhances the extraction of structural and content-related details from tables, improving perception and comprehension tasks.

Frozen vs. Unfrozen Low-Resolution Vision Encoder. We further investigate different training strategies in terms of the low-resolution vision encoder. As shown in Tab. 10, it is observed that no significant performance improvement but with longer training time consumption by unfreezing it, which is in line with the conclusion in the pioneering work [97]. Besides, we suppose the encoder frozen can serve as a regularization, facilitating the extraction of layout information and alleviating potential overfitting problems, as well as more stable training. To strike the trade-off between computational consumption and performance, we thus freeze the low-resolution vision encoder during training.

6 Limitation

In this section, we discuss the limitations of our TabPedia. Firstly, since we represent the table structure with regular rectangular boxes, TabPedia is currently not capable of accurately parsing structural information for twisted or distorted tables. Secondly, all images in TQA datasets, including WTQ [82], TabFact [83] and ComTQA are dominated by tables. Therefore, TabPedia still lacks the capability to directly answer the table question with original document image. In addition, compared to parallel decoding algorithms such as DETR [56] and Faster R-CNN [85], it consumes longer decoding time. Meantime, certain algorithmic designs such as KV cache, flash attention, and hardware improvements can effectively improve inference efficiency. We believe that with the iterative development of large model technology, the inference efficiency of TabPedia can be significantly improved.

7 Conclusion

In this paper, we propose a novel large vision-language model to unify diverse visual table understanding tasks, namely TabPedia. Specifically, we present a *concept synergy* mechanism to seamlessly integrate diverse tasks and multi-source visual tokens embedded from dual vision encoders as *concepts*. This mechanism is implemented by introducing the *meditative tokens* into the LLM. Then, we fully leverage the capability of LLMs to effectively understand these concepts and generate accurate and plausible responses. Extensive quantitative and qualitative experiments across various public benchmarks validate the effectiveness of our TabPedia. To further investigate the potential of TabPedia, we establish a challenging table VQA dataset, ComTQA, featuring round 9,000 QA pairs.

References

[1] Liangcai Gao, Yilun Huang, Herve Dejean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Lang. Icdar 2019 competition on table detection and recognition (ctdar). In

International Conference on Document Analysis and Recognition, 2019.

[2] Max Gobel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. Icdar 2013 table competition. In *International Conference on Document Analysis and Recognition*, 2013.

[3] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 572–573, 2020.

[4] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *international conference on document analysis and recognition*, pages 1162–1167, 2017.

[5] Xinyi Zheng, Douglas Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 697–706, 2021.

[6] Shoaib Ahmed Siddiqui, Muhammad Imran Malik, Stefan Agne, Andreas Dengel, and Sheraz Ahmed. DeCNT: Deep deformable cnn for table detection. *IEEE access*, 6:74151–74161, 2018.

[7] Duc-Dung Nguyen. TableSegNet: a fully convolutional network for table detection and segmentation in document images. *International Journal on Document Analysis and Recognition*, 25(1):1–14, 2022.

[8] Daqian Zhang, Ruibin Mao, Runting Guo, Yang Jiang, and Jing Zhu. YOLO-Table: disclosure document table detection with involution. *International Journal on Document Analysis and Recognition*, 26(1):1–14, 2023.

[9] Brandon Smock, Rohith Pesala, and Robin Abraham. PubTables-1M: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4634–4642, 2022.

[10] Hao Feng, Qi Liu, Hao Liu, Wengang Zhou, Houqiang Li, and Can Huang. Docpedia: Unleashing the power of large multimodal model in the frequency domain for versatile document understanding. *arXiv preprint arXiv:2311.11810*, 2023.

[11] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[12] Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. Textmonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*, 2024.

[13] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, et al. mPLUG-DocOwl: Modularized multimodal large language model for document understanding. *arXiv:2307.02499*, 2023.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations*, 2020.

[15] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763, 2021.

[16] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10012–10022, 2021.

[18] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

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

[20] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[21] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proceedings of the Advances in neural information processing systems*, pages 1877–1901, 2020.

[22] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[23] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Proceedings of the Advances in neural information processing systems*, pages 23716–23736, 2022.

[24] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv:2209.06794*, 2022.

[25] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv:2301.12597*, 2023.

[26] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. *arXiv:2305.06500*, 2023.

[27] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv:2302.14045*, 2023.

[28] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv:2306.14824*, 2023.

[29] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. *arXiv:2304.10592*, 2023.

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[31] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mPLUG-Owl: Modularization empowers large language models with multimodality. *arXiv:2304.14178*, 2023.

[32] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv:2306.15195*, 2023.

[33] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv:2305.03726*, 2023.

[34] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023.

[35] OpenAI. Gpt-4 technical report, 2023.

[36] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proceedings of the Advances in neural information processing systems*, 2020.

[37] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv:2305.10403*, 2023.

[38] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv:2304.15010*, 2023.

[39] Qwen. Introducing qwen-7b: Open foundation and human-aligned models (of the state-of-the-arts), 2023. URL `https://github.com/QwenLM/Qwen-7B`.

[40] Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Chen Li, Ji Zhang, Qin Jin, Fei Huang, et al. mPLUG-DocOwl 1.5: Unified structure learning for ocr-free document understanding. *arXiv preprint arXiv:2403.12895*, 2024.

[41] Jianqiang Wan, Sibo Song, Wenwen Yu, Yuliang Liu, Wenqing Cheng, Fei Huang, Xiang Bai, Cong Yao, and Zhibo Yang. OmniParser: A unified framework for text spotting, key information extraction and table recognition. *arXiv preprint arXiv:2403.19128*, 2024.

[42] ShengYun Peng, Seongmin Lee, Xiaojing Wang, Rajarajeswari Balasubramaniyan, and Duen Horng Chau. UniTable: Towards a unified framework for table structure recognition via self-supervised pretraining. *arXiv preprint arXiv:2403.04822*, 2024.

[43] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. OCR-free document understanding transformer. In *Proceedings of the European Conference on Computer Vision*, pages 498–517, 2022.

[44] Thomas Kieninger and Andreas Dengel. The t-recs table recognition and analysis system. In *International Association on Pattern Recognition*, pages 255–270, 1999.

[45] Basilios Gatos, Dimitrios Danatsas, Ioannis Pratikakis, and Stavros J Perantonis. Automatic table detection in document images. In *International Conference on Advances in Pattern Recognition*, pages 609–618, 2005.

[46] Gaurav Harit and Anukriti Bansal. Table detection in document images using header and trailer patterns. In *Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing*, pages 1–8, 2012.

[47] Nguyen D Vo, Khanh Nguyen, Tam V Nguyen, and Khang Nguyen. Ensemble of deep object detectors for page object detection. In *Proceedings of the International Conference on Ubiquitous Information Management and Communication*, pages 1–6, 2018.

[48] Azka Gilani, Shah Rukh Qasim, Imran Malik, and Faisal Shafait. Table detection using deep learning. In *international conference on document analysis and recognition*, volume 1, pages 771–776, 2017.

[49] Yilun Huang, Qinqin Yan, Yibo Li, Yifan Chen, Xiong Wang, Liangcai Gao, and Zhi Tang. A yolo-based table detection method. In *International Conference on Document Analysis and Recognition*, pages 813–818, 2019.

[50] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 572–573, 2020.

[51] Madhav Agarwal, Ajoy Mondal, and CV Jawahar. Cdec-net: Composite deformable cascade network for table detection in document images. In *international conference on pattern recognition*, pages 9491–9498, 2021.

[52] Ningning Sun, Yuanping Zhu, and Xiaoming Hu. Faster r-cnn based table detection combining corner locating. In *international conference on document analysis and recognition*, pages 1314–1319, 2019.

[53] Pau Riba, Anjan Dutta, Lutz Goldmann, Alicia Fornés, Oriol Ramos, and Josep Lladós. Table detection in invoice documents by graph neural networks. In *International Conference on Document Analysis and Recognition*, pages 122–127, 2019.

[54] Martin Holeček, Antonín Hoskovec, Petr Baudiš, and Pavel Klinger. Table understanding in structured documents. In *International Conference on Document Analysis and Recognition Workshops*, pages 158–164, 2019.

[55] Minghao Li, Yiheng Xu, Lei Cui, Shaohan Huang, Furu Wei, Zhoujun Li, and Ming Zhou. Docbank: A benchmark dataset for document layout analysis. In *Annual Meeting of the Association for Computational Linguistics*, pages 949–960, 2020.

[56] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020.

[57] Zengyuan Guo, Yuechen Yu, Pengyuan Lv, Chengquan Zhang, Haojie Li, Zhihui Wang, Kun Yao, Jingtuo Liu, and Jingdong Wang. Trust: an accurate and end-to-end table structure recognizer using splitting-based transformers. *arXiv preprint arXiv:2208.14687*, 2022.

[58] Weihong Lin, Zheng Sun, Chixiang Ma, Mingze Li, Jiawei Wang, Lei Sun, and Qiang Huo. TSRFormer: Table structure recognition with transformers. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6473–6482, 2022.

[59] Hao Liu, Xin Li, Mingming Gong, Bing Liu, Yunfei Wu, Deqiang Jiang, Yinsong Liu, and Xing Sun. Grab what you need: Rethinking complex table structure recognition with flexible components deliberation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3603–3611, 2024.

[60] Shubham Singh Paliwal, D Vishwanath, Rohit Rahul, Monika Sharma, and Lovekesh Vig. TableNet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In *International Conference on Document Analysis and Recognition*, pages 128–133, 2019.

[61] Hao Liu, Xin Li, Bing Liu, Deqiang Jiang, Yinsong Liu, Bo Ren, and Rongrong Ji. Show, read and reason. In *Proceedings of the ACM International Conference on Multimedia*, 2021.

[62] Zewen Chi, Heyan Huang, Heng-Da Xu, Houjin Yu, Wanxuan Yin, and Xian-Ling Mao. Complicated table structure recognition. *arXiv preprint arXiv:1908.04729*, 2019.

[63] Hao Liu, Xin Li, Bing Liu, Deqiang Jiang, Yinsong Liu, and Bo Ren. Neural collaborative graph machines for table structure recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4533–4542, 2022.

[64] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: data, model, and evaluation. In *Proceedings of the European Conference on Computer Vision*, pages 564–580, 2020.

[65] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[66] Hao Feng, Zijian Wang, Jingqun Tang, Jinghui Lu, Wengang Zhou, Houqiang Li, and Can Huang. Unidoc: A universal large multimodal model for simultaneous text detection, recognition, spotting and understanding. *arXiv preprint arXiv:2308.11592*, 2023.

[67] Haoran Wei, Lingyu Kong, Jinyue Chen, Liang Zhao, Zheng Ge, Jinrong Yang, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Vary: Scaling up the vision vocabulary for large vision-language models. *arXiv preprint arXiv:2312.06109*, 2023.

[68] Masato Fujitake. LayoutLLM: Large language model instruction tuning for visually rich document understanding. *arXiv preprint arXiv:2403.14252*, 2024.

[69] Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *Proceedings of the International Conference on Learning Representations*, 2023.

[70] Fuzhao Xue, Valerii Likhosherstov, Anurag Arnab, Neil Houlsby, Mostafa Dehghani, and Yang You. Adaptive computation with elastic input sequence. In *International Conference on Machine Learning*, pages 38971–38988, 2023.

[71] Mikhail S Burtsev, Yuri Kuratov, Anton Peganov, and Grigory V Sapunov. Memory transformer. *arXiv preprint arXiv:2006.11527*, 2020.

[72] Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. *Proceedings of the Advances in neural information processing systems*, 35:11079–11091, 2022.

[73] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *Proceedings of the International Conference on Learning Representations*, 2023.

[74] Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. *arXiv preprint arXiv:2310.02226*, 2023.

[75] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023.

[76] Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912, 2023.

[77] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Guohai Xu, Chenliang Li, Junfeng Tian, Qi Qian, Ji Zhang, et al. UReader: Universal ocr-free visually-situated language understanding with multimodal large language model. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2841–2858, 2023.

[78] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11474–11481, 2020.

[79] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1457–1464, 2011.

[80] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5676–5685, 2018.

[81] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys*, 51(6):1–36, 2019.

[82] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Annual Meeting of the Association for Computational Linguistics*, pages 1470–1480, 2015.

[83] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. TabFact: A large-scale dataset for table-based fact verification. In *Proceedings of the International Conference on Learning Representations*, 2019.

[84] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[85] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Proceedings of the Advances in neural information processing systems*, 28, 2015.

[86] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[87] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[88] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, pages 369–386, 2019.

[89] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the International Conference on Learning Representations*, 2018.

[90] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Proceedings of the Advances in neural information processing systems*, 32, 2019.

[91] Brandon Smock, Rohith Pesala, and Robin Abraham. GriTS: Grid table similarity metric for table structure recognition. In *International Conference on Document Analysis and Recognition*, pages 535–549, 2023.

[92] Yuliang Liu, Zhang Li, Hongliang Li, Wenwen Yu, Mingxin Huang, Dezhi Peng, Mingyu Liu, Mingrui Chen, Chunyuan Li, Lianwen Jin, et al. On the hidden mystery of ocr in large multimodal models. *arXiv preprint arXiv:2305.07895*, 2023.

[93] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607*, 2023.

[94] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.

[95] GPT-4V(ision) system card. 2023.

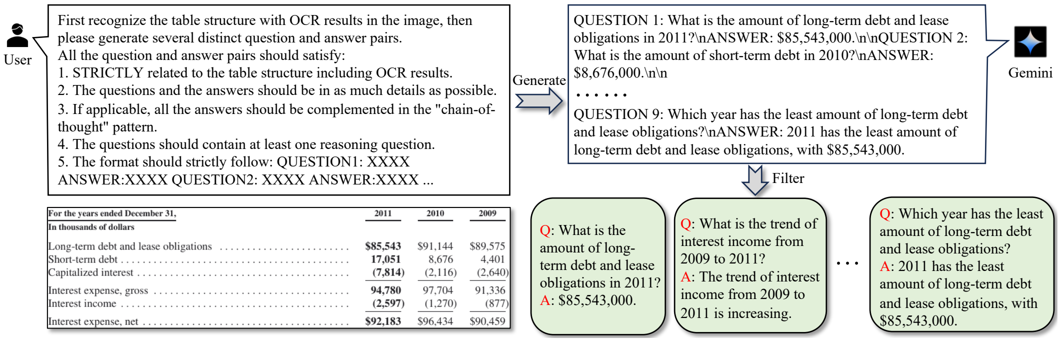
[96] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, et al. InternLM-XComposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*, 2024.

[97] Xiaohu Huang, Hao Zhou, Kun Yao, and Kai Han. Froster: Frozen clip is a strong teacher for open-vocabulary action recognition. In *Proceedings of the International Conference on Learning Representations*.

A More details about TQA datasets

A.1 QA Pairs Generation

We depict the procedure of collecting QA pairs with an example in Fig. A1. For input image, Gemini Pro [87] is prompted to first recognize the table structure with OCR results in the image, then generate several question and answer pairs according to OCR results. In order to improve the reliability of the generated answers, we leverage various prompting techniques, *i.e,* Chain-of-Thought and few-shot prompting. According to the specific prompt, Gemini Pro will generate multiple QA pairs for each input image and return them in an agreed-upon format. After obtaining raw responses generated by Gemini Pro, we utilize the regularized matching algorithm and the special character filter in turn to extract available question and answer pairs.



First recognize the table structure with OCR results in the image, then please generate several distinct question and answer pairs.
All the question and answer pairs should satisfy:
1. STRICTLY related to the table structure including OCR results.
2. The questions and the answers should be in as much details as possible.
3. If applicable, all the answers should be complemented in the "chain-of-thought" pattern.
4. The questions should contain at least one reasoning question.
5. The format should strictly follow: QUESTION1: XXXX ANSWER:XXXX QUESTION2: XXXX ANSWER:XXXX ...

For the years ended December 31, In thousands of dollars	2011	2010	2009
Long-term debt and lease obligations	**$85,543**	$91,144	$89,575
Short-term debt	**17,051**	8,676	4,401
Capitalized interest	**(7,814)**	(2,116)	(2,640)
Interest expense, gross	**94,780**	97,704	91,336
Interest income	**(2,597)**	(1,270)	(877)
Interest expense, net	**$92,183**	$96,434	$90,459

QUESTION 1: What is the amount of long-term debt and lease obligations in 2011?\nANSWER: $85,543,000.\n\nQUESTION 2: What is the amount of short-term debt in 2010?\nANSWER: $8,676,000.\n\n
......
QUESTION 9: Which year has the least amount of long-term debt and lease obligations?\nANSWER: 2011 has the least amount of long-term debt and lease obligations, with $85,543,000.

Q: What is the amount of long-term debt and lease obligations in 2011?
A: $85,543,000.

Q: What is the trend of interest income from 2009 to 2011?
A: The trend of interest income from 2009 to 2011 is increasing.

...

Q: Which year has the least amount of long-term debt and lease obligations?
A: 2011 has the least amount of long-term debt and lease obligations, with $85,543,000.

Figure A1: The illustration of an example for generating QA pairs with the powerful LVLM, Gemini Pro [87]. The prompt includes several key rules to ensure the response quality as much as possible.

A.2 ComTQA Benchmark

In Tab. A1, we present the distribution of both data sources [5, 9] within the ComTQA dataset. Concretely, ComTQA comprises a total of 9,070 QA pairs across 1,591 images, averaging 5 questions per image. Different from existing TQA benchmarks [82, 83], ComTQA contains more complex table questions in real-world table images to assess the robustness of various models. As shown in Fig. A2, we showcase several representative examples, including multiple answers, mathematical calculation and logical inference, which are the question types lacking in previous benchmarks. To this end, we hope that ComTQA could fill this gap and serve as a reasonable benchmark for community development.

Table A1: Statistics of ComTQA benchmark.

	PubTab1M	FinTabNet	Total
#images	932	659	1,591
#QA pairs	6,232	2,838	9,070
Avg. per image	6	4	5

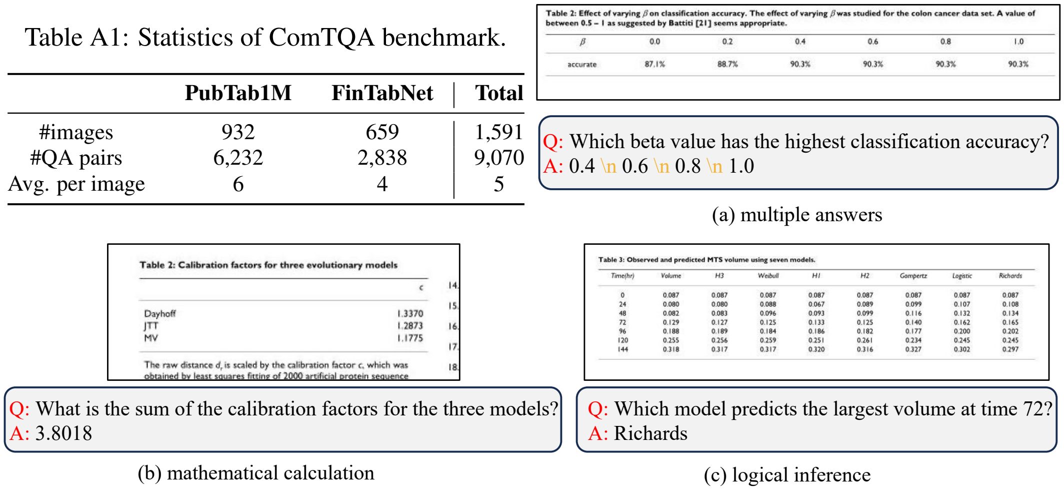

14.
15.
16.
17.
18.

Table 2: Effect of varying β on classification accuracy. The effect of varying β was studied for the colon cancer data set. A value of between 0.5 – 1 as suggested by Battiti [21] seems appropriate.

β	0.0	0.2	0.4	0.6	0.8	1.0
accurate	87.1%	88.7%	90.3%	90.3%	90.3%	90.3%

Q: Which beta value has the highest classification accuracy?
A: 0.4 \n 0.6 \n 0.8 \n 1.0

(a) multiple answers

Table 2: Calibration factors for three evolutionary models

	c
Dayhoff	1.3370
JTT	1.2873
MV	1.1775

The raw distance d_r is scaled by the calibration factor c, which was obtained by least squares fitting of 2000 artificial protein sequence

Q: What is the sum of the calibration factors for the three models?
A: 3.8018

(b) mathematical calculation

Table 3: Observed and predicted MTS volume using seven models.

Time(hr)	*Volume*	*H3*	*Weibull*	*H1*	*H2*	*Gompertz*	*Logistic*	*Richards*
0	0.087	0.087	0.087	0.087	0.087	0.087	0.087	0.087
24	0.080	0.080	0.088	0.067	0.089	0.099	0.107	0.108
48	0.082	0.083	0.096	0.093	0.099	0.116	0.132	0.134
72	0.129	0.127	0.125	0.133	0.125	0.140	0.162	0.165
96	0.188	0.189	0.184	0.186	0.182	0.177	0.200	0.202
120	0.255	0.256	0.259	0.251	0.261	0.234	0.245	0.245
144	0.318	0.317	0.317	0.320	0.316	0.327	0.302	0.297

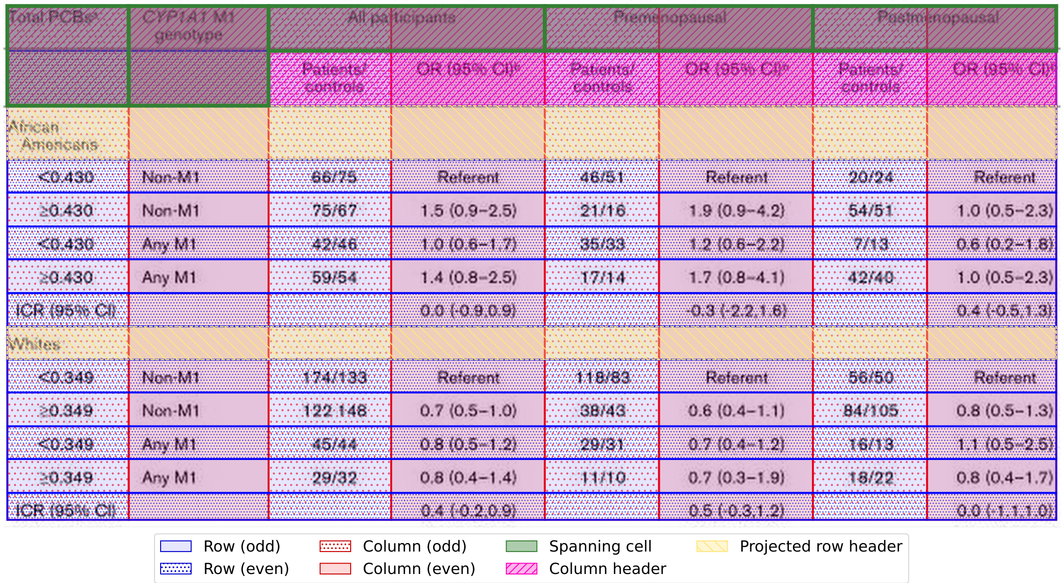
Q: Which model predicts the largest volume at time 72?
A: Richards

(c) logical inference

Figure A2: More visualization on ComTQA benchmark. We display several complex QA types, such as multiple answers, mathematical calculation and logical inference. Zoom in for best view.

B Annotation in TSR task

We illustrate the object classes utilized in TSR and TQ tasks as shown in Fig. B3. A table generally is composed of five basic elements, i.e., column, row, spanning cell, column header and projected row header. "Row" denotes the rectangular boxes of each row's content in the table, while "Column" denotes the rectangular boxes of each column's content. The area where each row and each column intersect represents the table cell. Besides these both most common table elements, "Column header" refers to the area in the table that contains the data type or content for each column, usually occupying multiple rows at the top of the table. "Projected row header", as a special row, represents the area that contains a single non-blank cell in a row. "Spanning cell" refers to a cell in a table that spans multiple rows or columns. According to these definitions, these objects have implicit relationship and construct a table's hierarchical structure through physically overlapped rectangle boxes



Total PCBs[a]	CYP1A1 M1 genotype	All participants		Premenopausal		Postmenopausal	
		Patients/ controls	OR (95% CI)[b]	Patients/ controls	OR (95% CI)[b]	Patients/ controls	OR (95% CI)[b]
African Americans							
<0.430	Non-M1	66/75	Referent	46/51	Referent	20/24	Referent
≥0.430	Non-M1	75/67	1.5 (0.9–2.5)	21/16	1.9 (0.9–4.2)	54/51	1.0 (0.5–2.3)
<0.430	Any M1	42/46	1.0 (0.6–1.7)	35/33	1.2 (0.6–2.2)	7/13	0.6 (0.2–1.8)
≥0.430	Any M1	59/54	1.4 (0.8–2.5)	17/14	1.7 (0.8–4.1)	42/40	1.0 (0.5–2.3)
ICR (95% CI)			0.0 (-0.9,0.9)		-0.3 (-2.2,1.6)		0.4 (-0.5,1.3)
Whites							
<0.349	Non-M1	174/133	Referent	118/83	Referent	56/50	Referent
≥0.349	Non-M1	122 148	0.7 (0.5–1.0)	38/43	0.6 (0.4–1.1)	84/105	0.8 (0.5–1.3)
<0.349	Any M1	45/44	0.8 (0.5–1.2)	29/31	0.7 (0.4–1.2)	16/13	1.1 (0.5–2.5)
≥0.349	Any M1	29/32	0.8 (0.4–1.4)	11/10	0.7 (0.3–1.9)	18/22	0.8 (0.4–1.7)
ICR (95% CI)			0.4 (-0.2,0.9)		0.5 (-0.3,1.2)		0.0 (-1.1,1.0)

Figure B3: The illustration of an example table with dilated bounding box annotations for different object classes for modeling table structure recognition.

C Broader Impact

Our proposed model targets to unify multiple visual form comprehension tasks. This technology could help more people with visual impairments access tabular data through cooperating with improved screen readers and other assistive technologies. Moreover, automating table understanding technology could reduce the need for time-consuming manual data entry and correction, freeing up human resources for more complex and creative tasks. To be honest, this technology also brings some negative societal impacts. As more table data is extracted and processed with automatic visual table understanding, there is a heightened risk of sensitive information being mishandled or exposed. It is crucial to ensure robust data privacy measures.

D More Qualitative Results

Results on in-the-wild cases. For better investigating the generalization of our proposed TabPedia, we randomly select some document images from a document website and illustrate the generation results in Fig. D5. For perception and comprehension tasks, TabPedia generates accurate and reasonable responses in TD, TSR and TQA tasks, which sufficiently proves the robustness of our method for visual table understanding.

Attention map of meditative tokens. In order to analyze the information extraction of meditative tokens for different tasks, we visualized the attention maps of meditative tokens for input instructions with different granularity of visual feature tokens, as shown in Fig. D4. For each task, we select the shallow and deep four-layer attention maps in the LLM for visualization, respectively. The y-axis represents the meditative tokens, while the x-axis represents the sequence of instruction tokens and different granular visual tokens. For perceptive tasks, meditative tokens are densely attentive to most of the input information in the shallow layers, while they showcase diverse attention regions in the deeper layers.This phenomenon illustrates that meditative tokens could adaptively capture task-related information with respect to diverse tasks. For the comprehension task (TQA), meditative tokens show a different attention pattern from perception tasks, which maintain sparse attention with input tokens in the shallow layers. These results validate that our proposed meditative tokens adaptively enable different regions of visual tokens and understand the intention of specific task questions.

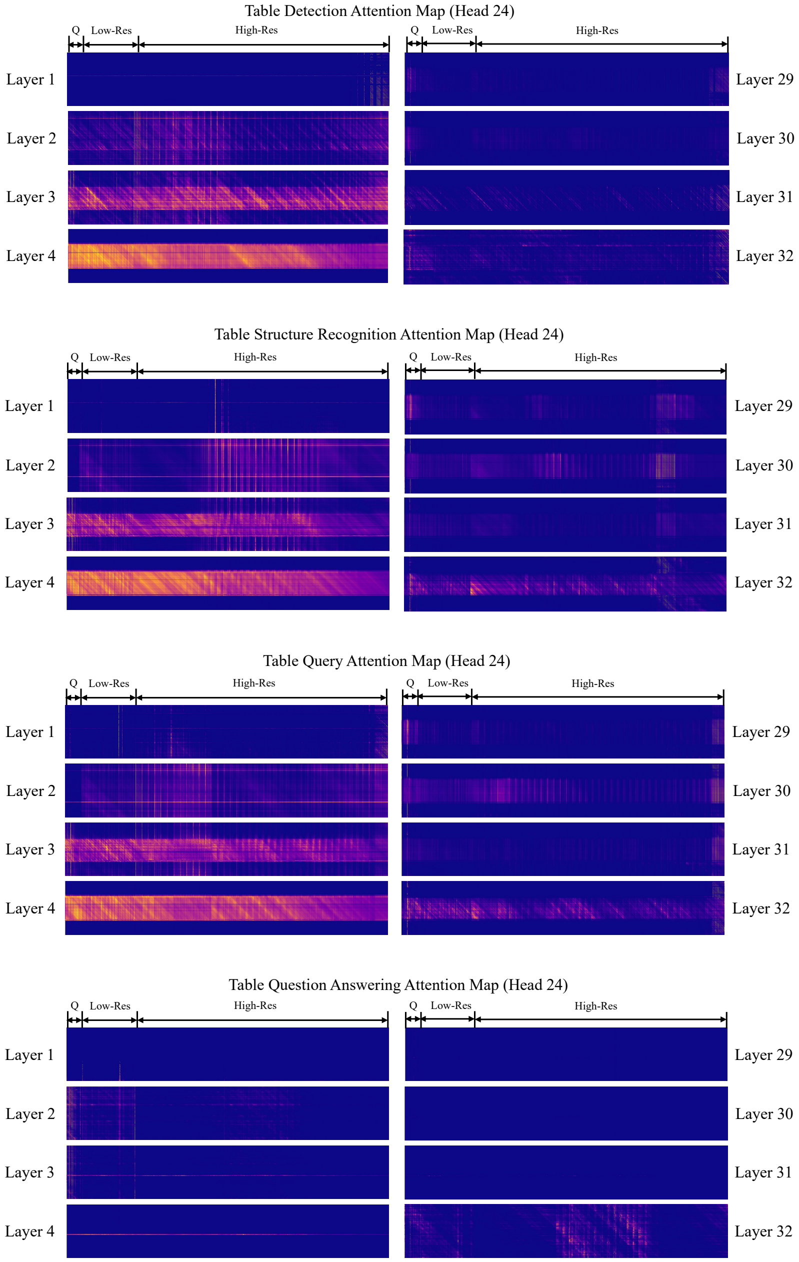
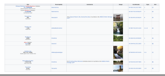
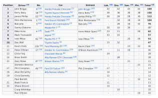
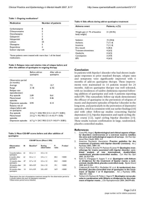
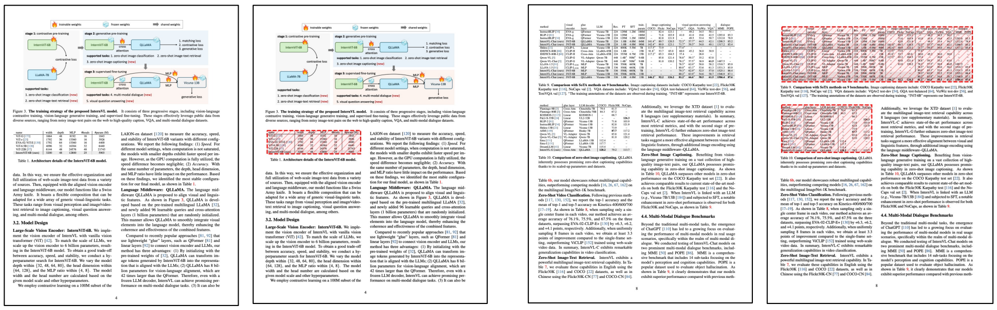


Figure D4: Visualization of attention maps between meditative tokens and the sequence of instruction and visual tokens. “Q”, “Low-Res” and “High-Res” denote the instruction tokens, global visual tokens and local visual tokens, respectively. Y-axis denotes the meditative tokens. Zoom in for best view.

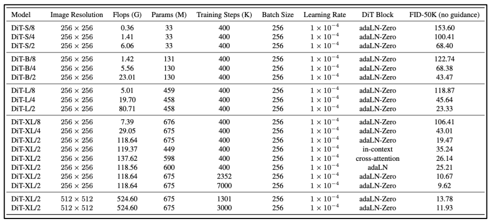
Visualization of TabPedia's responses. As shown in Tab. D2,we introducing meditative tokens can bring promising performance across VTU tasks. We compare in detail the differences in the generated results before and after the introduction of Meditative in different VTU tasks. It is observed that introducing meditative tokens mainly improves the quality of long-form responses. Also for the perception tasks including TD and TSR, introducing meditative tokens can alleviate the meaningless or repetitive word generation. For the comprehension task, TQA, introducing meditative tokens can generate more elaborated and reasonable response. As suggested, we showcase several samples for better understanding.

Table D2: Qualitative results of TabPedia's responses.

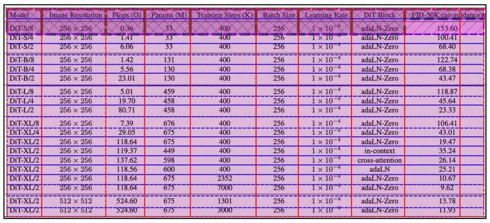
Image	**Question**	**GT**	**TabPedia (w/o Meditative Tokens)**	**TabPedia**
	what is the last site's municipality?	"Fukuyama"	"Fukuoka \n Reason: The last site is Tachibana, and its municipality is Fukuoka."	"Fukuyama \nReason: It is shown in the last row of the table that the last site's municipality is Fukuyama. So the answer is Fukuyama."
	john briggs and james phillip both drove which type of car?	"Honda Prelude Chevrolet"	"Mazda RX-7 \n Reason: From the image, we can see that both John Briggs and James Phillipps drove Mazda RX-7."	"Honda Prelude Chevrolet \n Reason: From the image, we can see that both John Briggs and James Phillip drove the Honda Prelude Chevrolet. John Briggs drove the No.9 Honda Prelude Chevrolet, while James Phillip drove the No.10 Honda Prelude Chevrolet. So they both drove the same type of car, which is the Honda Prelude Chevrolet."
	Output all the table element's locations in the photo.	"Here is a list of all the locations of table element in the picture:\n [0.095,0.139,0.424,0.279]\n [0.095,0.375,0.458,0.620]\n [0.092,0.704,0.472,0.862]\n [0.518,0.155,0.807,0.321]"	"Here is a list of all the locations of table element in the picture:\n [0.095,0.139,0.426,0.329]\n [0.095,0.409,0.447,0.669]\n [0.095,0.699,0.459,0.859]"	"Here is a list of all the locations of table element in the picture:\n [0.096,0.140,0.422,0.281]\n [0.095,0.378,0.456,0.617]\n [0.094,0.707,0.474,0.862]\n [0.518,0.156,0.809,0.324]"



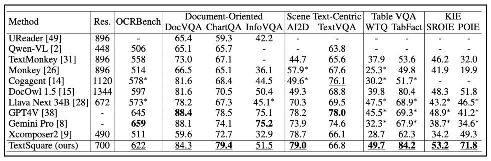
(a) In-the-wild cases on TD task



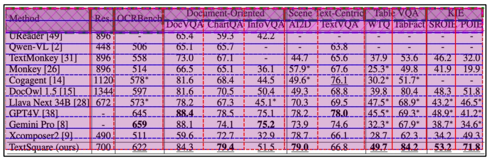
Model	Image Resolution	Flops (G)	Params (M)	Training Steps (K)	Batch Size	Learning Rate	DiT Block	FID-50K (no guidance)
DiT-S/8	256 × 256	0.36	33	400	256	1×10^{-4}	adaLN-Zero	153.60
DiT-S/4	256 × 256	1.41	33	400	256	1×10^{-4}	adaLN-Zero	100.41
DiT-S/2	256 × 256	6.06	33	400	256	1×10^{-4}	adaLN-Zero	68.40
DiT-B/8	256 × 256	1.42	131	400	256	1×10^{-4}	adaLN-Zero	122.74
DiT-B/4	256 × 256	5.56	130	400	256	1×10^{-4}	adaLN-Zero	68.38
DiT-B/2	256 × 256	23.01	130	400	256	1×10^{-4}	adaLN-Zero	43.47
DiT-L/8	256 × 256	5.01	459	400	256	1×10^{-4}	adaLN-Zero	118.87
DiT-L/4	256 × 256	19.70	458	400	256	1×10^{-4}	adaLN-Zero	45.64
DiT-L/2	256 × 256	80.71	458	400	256	1×10^{-4}	adaLN-Zero	23.33
DiT-XL/8	256 × 256	7.39	676	400	256	1×10^{-4}	adaLN-Zero	106.41
DiT-XL/4	256 × 256	29.05	675	400	256	1×10^{-4}	adaLN-Zero	43.01
DiT-XL/2	256 × 256	118.64	675	400	256	1×10^{-4}	adaLN-Zero	19.47
DiT-XL/2	256 × 256	119.37	449	400	256	1×10^{-4}	in-context	35.24
DiT-XL/2	256 × 256	137.62	598	400	256	1×10^{-4}	cross-attention	26.14
DiT-XL/2	256 × 256	118.56	600	400	256	1×10^{-4}	adaLN	25.21
DiT-XL/2	256 × 256	118.64	675	2352	256	1×10^{-4}	adaLN-Zero	10.67
DiT-XL/2	256 × 256	118.64	675	7000	256	1×10^{-4}	adaLN-Zero	9.62
DiT-XL/2	512 × 512	524.60	675	1301	256	1×10^{-4}	adaLN-Zero	13.78
DiT-XL/2	512 × 512	524.60	675	3000	256	1×10^{-4}	adaLN-Zero	11.93



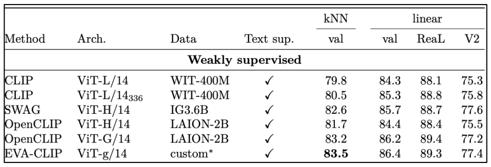
Model	Image Resolution	Flops (G)	Params (M)	Training Steps (K)	Batch Size	Learning Rate	DiT Block	FID-50K (no guidance)
DiT-S/8	256 × 256	0.36	33	400	256	1×10^{-4}	adaLN-Zero	153.60
DiT-S/4	256 × 256	1.41	33	400	256	1×10^{-4}	adaLN-Zero	100.41
DiT-S/2	256 × 256	6.06	33	400	256	1×10^{-4}	adaLN-Zero	68.40
DiT-B/8	256 × 256	1.42	131	400	256	1×10^{-4}	adaLN-Zero	122.74
DiT-B/4	256 × 256	5.56	130	400	256	1×10^{-4}	adaLN-Zero	68.38
DiT-B/2	256 × 256	23.01	130	400	256	1×10^{-4}	adaLN-Zero	43.47
DiT-L/8	256 × 256	5.01	459	400	256	1×10^{-4}	adaLN-Zero	118.87
DiT-L/4	256 × 256	19.70	458	400	256	1×10^{-4}	adaLN-Zero	45.64
DiT-L/2	256 × 256	80.71	458	400	256	1×10^{-4}	adaLN-Zero	23.33
DiT-XL/8	256 × 256	7.39	676	400	256	1×10^{-4}	adaLN-Zero	106.41
DiT-XL/4	256 × 256	29.05	675	400	256	1×10^{-4}	adaLN-Zero	43.01
DiT-XL/2	256 × 256	118.64	675	400	256	1×10^{-4}	adaLN-Zero	19.47
DiT-XL/2	256 × 256	119.37	449	400	256	1×10^{-4}	in-context	35.24
DiT-XL/2	256 × 256	137.62	598	400	256	1×10^{-4}	cross-attention	26.14
DiT-XL/2	256 × 256	118.56	600	400	256	1×10^{-4}	adaLN	25.21
DiT-XL/2	256 × 256	118.64	675	2352	256	1×10^{-4}	adaLN-Zero	10.67
DiT-XL/2	256 × 256	118.64	675	7000	256	1×10^{-4}	adaLN-Zero	9.62
DiT-XL/2	512 × 512	524.60	675	1301	256	1×10^{-4}	adaLN-Zero	13.78
DiT-XL/2	512 × 512	524.60	675	3000	256	1×10^{-4}	adaLN-Zero	11.93



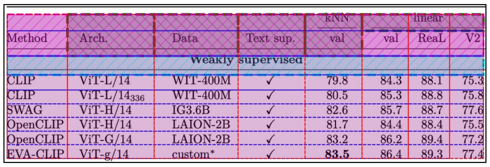
Method	Res.	OCRBench	Document-Oriented			Scene Text-Centric		Table VQA		KIE	
			DocVQA	ChartQA	InfoVQA	AI2D	TextVQA	WTQ	TabFact	SROIE	POIE
UReader [49]	896	-	65.4	59.3	42.2	-	-	-	-	-	-
Qwen-VL [2]	448	506	65.1	65.7	-	-	63.8	-	-	-	-
TextMonkey [31]	896	558	73.0	67.1	-	44.7	65.6	37.9	53.6	46.2	32.0
Monkey [26]	896	514	66.5	65.1	36.1	57.9*	67.6	25.3*	49.8	41.9	19.9
Cogagent [14]	1120	578*	81.6	68.4	44.5	49.6*	76.1	30.2*	51.7*	-	-
DocOwl 1.5 [15]	1344	597	81.6	70.5	50.4	49.3	68.8	39.8	80.4	48.3	51.8
Llava Next 34B [28]	672	573*	78.2	67.3	45.1*	70.3	69.5	47.5*	68.9*	43.2*	46.5*
GPT4V [38]	-	645	**88.4**	78.5	75.1	78.2	**78.0**	45.5*	69.3*	48.9*	41.2*
Gemini Pro [8]	-	**659**	88.1	74.1	**75.2**	73.9	74.6	32.3*	67.9*	38.7*	34.6*
Xcomposer2 [9]	490	511	59.6	72.7	32.9	78.7	66.1	28.7	62.3	34.2	49.3
TextSquare (ours)	700	622	84.3	**79.4**	51.5	**79.0**	66.8	**49.7**	**84.2**	**53.2**	**71.8**



Method	Res.	OCRBench	Document-Oriented			Scene Text-Centric		Table VQA		KIE	
			DocVQA	ChartQA	InfoVQA	AI2D	TextVQA	WTQ	TabFact	SROIE	POIE
UReader [49]	896	-	65.4	59.3	42.2	-	-	-	-	-	-
Qwen-VL [2]	448	506	65.1	65.7	-	-	63.8	-	-	-	-
TextMonkey [31]	896	558	73.0	67.1	-	44.7	65.6	37.9	53.6	46.2	32.0
Monkey [26]	896	514	66.5	65.1	36.1	57.9*	67.6	25.3*	49.8	41.9	19.9
Cogagent [14]	1120	578*	81.6	68.4	44.5	49.6*	76.1	30.2*	51.7*	-	-
DocOwl 1.5 [15]	1344	597	81.6	70.5	50.4	49.3	68.8	39.8	80.4	48.3	51.8
Llava Next 34B [28]	672	573*	78.2	67.3	45.1*	70.3	69.5	47.5*	68.9*	43.2*	46.5*
GPT4V [38]	-	645	**88.4**	78.5	75.1	78.2	**78.0**	45.5*	69.3*	48.9*	41.2*
Gemini Pro [8]	-	**659**	88.1	74.1	**75.2**	73.9	74.6	32.3*	67.9*	38.7*	34.6*
Xcomposer2 [9]	490	511	59.6	72.7	32.9	78.7	66.1	28.7	62.3	34.2	49.3
TextSquare (ours)	700	622	84.3	**79.4**	51.5	**79.0**	66.8	**49.7**	**84.2**	**53.2**	**71.8**

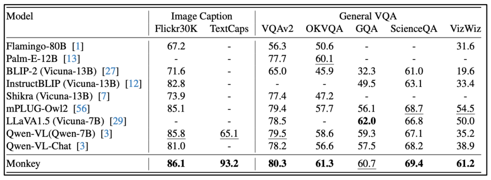


				kNN	linear		
Method	Arch.	Data	Text sup.	val	val	ReaL	V2
Weakly supervised							
CLIP	ViT-L/14	WIT-400M	✓	79.8	84.3	88.1	75.3
CLIP	ViT-L/14_{336}	WIT-400M	✓	80.5	85.3	88.8	75.8
SWAG	ViT-H/14	IG3.6B	✓	82.6	85.7	88.7	77.6
OpenCLIP	ViT-H/14	LAION-2B	✓	81.7	84.4	88.4	75.5
OpenCLIP	ViT-G/14	LAION-2B	✓	83.2	86.2	89.4	77.2
EVA-CLIP	ViT-g/14	custom*	✓	**83.5**	86.4	89.3	77.4



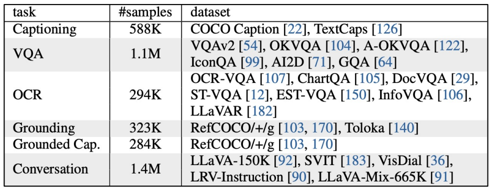
				kNN	linear		
Method	Arch.	Data	Text sup.	val	val	ReaL	V2
Weakly supervised							
CLIP	ViT-L/14	WIT-400M	✓	79.8	84.3	88.1	75.3
CLIP	ViT-L/14_{336}	WIT-400M	✓	80.5	85.3	88.8	75.8
SWAG	ViT-H/14	IG3.6B	✓	82.6	85.7	88.7	77.6
OpenCLIP	ViT-H/14	LAION-2B	✓	81.7	84.4	88.4	75.5
OpenCLIP	ViT-G/14	LAION-2B	✓	83.2	86.2	89.4	77.2
EVA-CLIP	ViT-g/14	custom*	✓	**83.5**	86.4	89.3	77.4

(b) In-the-wild cases on TSR task



Model	Image Caption		General VQA				
	Flickr30K	TextCaps	VQAv2	OKVQA	GQA	ScienceQA	VizWiz
Flamingo-80B [1]	67.2	-	56.3	50.6	-	-	31.6
Palm-E-12B [13]	-	-	77.7	60.1	-	-	-
BLIP-2 (Vicuna-13B) [27]	71.6	-	65.0	45.9	32.3	61.0	19.6
InstructBLIP (Vicuna-13B) [12]	82.8	-	-	-	49.5	63.1	33.4
Shikra (Vicuna-13B) [7]	73.9	-	77.4	47.2	-	-	-
mPLUG-Owl2 [56]	85.1	-	79.4	57.7	56.1	68.7	54.5
LLaVA1.5 (Vicuna-7B) [29]	-	-	78.5	-	**62.0**	66.8	50.0
Qwen-VL(Qwen-7B) [3]	85.8	65.1	79.5	58.6	59.3	67.1	35.2
Qwen-VL-Chat [3]	81.0	-	78.2	56.6	57.5	68.2	38.9
Monkey	**86.1**	**93.2**	**80.3**	**61.3**	60.7	**69.4**	**61.2**

Q: Answer the question and give a reason. what's the point of Flamingo-80B in VQAv2?
A: 56.3 Reason: Flamingo-80B is a model in VQAv2. Its VQAv2 score is shown in the second column, which is 56.3.



task	#samples	dataset
Captioning	588K	COCO Caption [22], TextCaps [126]
VQA	1.1M	VQAv2 [54], OKVQA [104], A-OKVQA [122], IconQA [99], AI2D [71], GQA [64]
OCR	294K	OCR-VQA [107], ChartQA [105], DocVQA [29], ST-VQA [12], EST-VQA [150], InfoVQA [106], LLaVAR [182]
Grounding	323K	RefCOCO/+/g [103, 170], Toloka [140]
Grounded Cap.	284K	RefCOCO/+/g [103, 170]
Conversation	1.4M	LLaVA-150K [92], SVIT [183], VisDial [36], LRV-Instruction [90], LLaVA-Mix-665K [91]

Q: Answer the question and give a reason. How many datasets utilized in the Conversation task?
A: 5 Reason: There are 5 datasets utilized in the Conversation task, which are LLaVA-150K, SVIT, VisDial, LLaVA-Mix-665K and LRV-Instruction.

(c) In-the-wild cases on TQA task

Figure D5: Qualitative results of TabPedia on in-the-wild cases. TabPedia achieves impressive performance in these unseen images, which validates its robustness and generalization. Zoom in for best view.