TableNet: A Large-Scale Table Dataset with LLM-Powered Autonomous Generation

Ruilin Zhang, Kai Yang*

School of Computer Science and Technology, Tongji University
4800 Cao'an Road, Shanghai, 201804, China

Abstract

Table Structure Recognition (TSR) requires the logical reasoning ability of large language models (LLMs) to handle complex table layouts, but current datasets are limited in scale and quality, hindering effective use of this reasoning capacity. We thus present **TableNet dataset**, a new table structure recognition dataset collected and generated through multiple sources. Central to our approach is the first **LLM-powered autonomous table generation and recognition multi-agent system** that we developed. The generation part of our system integrates controllable visual, structural, and semantic parameters into the synthesis of table images. It facilitates the creation of a wide array of semantically coherent tables, adaptable to user-defined configurations along with annotations, thereby supporting large-scale and detailed dataset construction. This capability enables a comprehensive and nuanced table image annotation taxonomy, potentially advancing research in table-related domains. In contrast to traditional data collection methods, This approach facilitates the theoretically infinite, domain-agnostic, and style-flexible generation of table images, ensuring both efficiency and precision. The recognition part of our system is a diversity-based active learning paradigm that utilizes tables from multiple sources and selectively samples most informative data to finetune a model, achieving a competitive performance on TableNet test set while reducing training samples by a large margin compared with baselines, and a much higher performance on web-crawled real-world tables compared with models trained on predominant table datasets. To the best of our knowledge, this is the first work which employs the concept of active learning into the structure recognition of tables which is diverse in numbers of rows or columns, merged cells, cell contents, etc, which fits better for diversity-based active learning.

Datasets — https://huggingface.co/datasets/AnonymousUser123123/TableNet/tree/main

Introduction

Table Structure Recognition (TSR) aims to recover the logical structure of tables from images. Despite recent advances, it remains challenging due to the wide variability in table layouts, styles, and semantics. Real-world tables frequently include complex visual patterns—merged cells, missing borders, inconsistent alignments, or heterogeneous color schemes, which brings challenges to the logical reasoning ability to large language models (LLM) when parse the structure of tables. However, existing datasets are often limited in scale, diversity, and annotation quality, making it hard to fully exploit the logical reasoning ability of LLMs.

To address these limitations, we introduce the TableNet dataset and the first LLM-powered autonomous table generation and recognition multi-agent system. By integrating controllable parameters, the system is capable of generating a diverse range of tables with minimal human intervention. Unlike traditional dataset collection methods, our system leverages prior knowledge embedded in LLMs and is user-configurable, enabling the generation scalable, controllable, semantically coherent tables and corresponding annotations.

The broad style and domain coverage of TableNet motivates the use of active learning, which selects maximally informative samples for model training and could be used for data filtering in many fields like intrusion/anomaly detection(Zhu and Yang 2019; Yang et al. 2018), computer vision(Beluch et al. 2018), etc. This procedure can be described using a five-step algorithm shown in Algorithm 1(Xia et al. 2025), that is initialize, query, annotate, train and stop. The existing active learning method can be categorized into two main streams: diversity-based and uncertainty-based. Diversity-based methods(Gissin and Shalev-Shwartz 2019; Sener and Savarese 2017) aims to select a subset representative of overall distribution. Uncertainty-based methods(Beluch et al. 2018; Gal, Islam, and Ghahramani 2017; Yoo and Kweon 2019) prioritize samples that model cannot predict for sure. Diversity-based active learning is particularly suitable for heterogeneous table structures. By actively selecting training instances from multiple sources, our TSR model achieves competitive performance on the TableNet test set while requiring substantially fewer samples and significantly outperforms models trained on existing datasets when evaluated on unseen real-world tables.

Contributions of this paper can summarized as follows.

- We release TableNet, a large-scale table dataset composed of synthetic tables generated by a controllable LLM-based multi-agent system, real-world tables collected from the web, and augmented open source dataset,

*Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

ensuring diversity in visual style, structure, and semantics.

- We develop the first LLM-based autonomous multi-agent system to generate table images with user-configurable properties, enabling synthesis of large-scale realistic dataset and serving as a fundamental support for TSR.
- We employ a diversity-based active learning strategy to train the table structure recognition part of our system. Training on actively selected samples, our model achieves a competitive performance with much less training samples on its test set and a higher performance than models trained on other public datasets on web-crawled unseen real-world tables.

Related Works

Table Structure Recognition (TSR). Tables are systematic arrangements of data organized into rows and columns, and their grid format provides clarity and efficiency. TSR aims to analyze the layout of table images and rebuild their cellular structure through representations such as markup languages (Zhong, ShafieiBavani, and Jimeno Yepes 2020; Nassar et al. 2022), spreadsheets (Koci et al. 2017), and graphs (Koci et al. 2019, 2018; Qasim, Mahmood, and Shafait 2019; Xue, Li, and Tao 2019; Chi et al. 2019; Liu et al. 2024). Over the years, TSR approaches have evolved rapidly, progressed through three phases: (1) early approaches utilizing rule-based and heuristic methods, (2) deep learning-based methods, and (3) LLM-based methods.

Early approaches from the 1990s to 2010s relied heavily on visual cues such as cell boundaries and alignments. Pyreddy et al. (Pyreddy and Croft 1997) introduced the Character Alignment Graph using blank-space patterns, while Rus et al. (Rus and Subramanian 1997) proposed a similar White Space Density Graph. Several other rule-based or heuristic methods followed (Itonori 1993; Kieninger and Dengel 1999; Shigarov, Mikhailov, and Altaev 2016). However, these techniques consistently struggled with complex table layouts, highlighting the need for more general solutions.

In the deep learning era, neural methods have dramatically surpassed early TSR approaches (Hashmi et al. 2021). Object detection–based methods (Zheng et al. 2021; Long et al. 2021; Schreiber et al. 2017; Raja, Mondal, and Jawahar 2020; Ajayi et al. 2024; Anand et al. 2023; Gorishniy et al. 2021) identify structural components such as rows, columns, and captions. Markup-language models (Zhong, ShafieiBavani, and Jimeno Yepes 2020; Nassar et al. 2022) treat TSR as image-to-markup translation using NLP-inspired architectures. Graph-based approaches (Koci et al. 2019, 2018; Qasim, Mahmood, and Shafait 2019; Xue, Li, and Tao 2019; Chi et al. 2019; Liu et al. 2024) represent tables as graphs to model cell-level relationships.

Recently, LLMs demonstrated exceptional ability in natural language related tasks. But LLM for TSR still remains an underexplored area and most work lies in table understanding field (Zheng et al. 2024; Li et al. 2023; Zha et al. 2023; Su et al. 2024; Pang et al. 2024; Sui et al. 2024). Zhou et al. (Zhou et al. 2024) proposed Neighbor-Guided Toolchain Reasoner framework and significantly enhanced the recognition capabilities of vanilla vision LLMs. The availability of more datasets is likely to spur a rapid proliferation of LLM-based methods, fully leveraging the potential of large language models in handling tabular data.

TSR Datasets. Progress in Table Structure Recognition (TSR) has been largely driven by the release of benchmark datasets. The ICDAR series (Göbel et al. 2013; Gao et al. 2017, 2019; Kayal et al. 2021) provides image-based datasets covering table detection, structure recognition, and end-to-end extraction, with table images ranging from PDFs to handwritten documents.

IBM Research has contributed several influential datasets. PubTabNet (Zhong, ShafieiBavani, and Jimeno Yepes 2020) focuses on TSR by converting table regions in PMCOA PDFs into HTML using the PubLayNet pipeline (Zhong, Tang, and Yepes 2019). FinTabNet (Zheng et al. 2021) addresses the limitation of PubTabNet, which only supports TSR and lacks full-page context and overall table region bounding boxes. Furthermore, SynthTabNet (Nassar et al. 2022) further synthesizes PubTabNet, FinTabNet, and TableBank (Li et al. 2020b) to overcome domain limitations and structural simplicity.

TableBank(Li et al. 2020b) is another large-scale TSR dataset. It is created using weak supervision by automatically extracting tables from Word and LaTeX documents collected from the internet. This method leverages the structural information already embedded in these formats, allowing the dataset to be built at scale without extensive manual labeling. As a result, TableBank supports better generalization in TSR models, making them more effective in handling diverse and real-world table layouts.

As shown in Table 1, existing datasets still suffer from limited collection pipelines, domain constraints, and insufficient diversity. Models trained on one dataset often generalize poorly to others due to the diverse visual nature of tables (Li et al. 2020a; Somvanshi et al. 2024). For example, a model trained on monochrome tables may perform poorly when applied to colored tables. Classic datasets such as UNLV (Cesarini et al. 2002) and Marmot (Fang et al. 2012), along with recent ones like PubTables-1M (Yang et al. 2023) and TabRecSet (Smock, Pesala, and Abraham 2022), either lack the diversity (Zheng et al. 2021; Zhong, ShafieiBavani, and Jimeno Yepes 2020; Nassar et al. 2022; Li et al. 2020b; Cesarini et al. 2002; Fang et al. 2012; Yang et al. 2023) or omit explicit table-type labels. (Long et al. 2021; Smock, Pesala, and Abraham 2022). To address these limitations, we released TableNet, a large-scale synthesized TSR dataset, together with the first autonomous table generation multi-agent system to potentially facilitate TSR.

Text-only LLM Image Synthesis. Recent works have explored leveraging text-only LLMs to synthetic images for VLMs because of the lack of vision-language data (Cascante-Bonilla et al. 2022; Johnson-Roberson et al. 2016) and this approach has been applied to the generation of QA pairs of charts, plots. (Kahou et al. 2017; Kafle et al. 2018) used small set of chart and fixed QA templates to generate chart VQA data. (Carbune et al. 2024; Li and Tajbakhsh 2023) used text-only LLMs to generate annota-

Table 1: **Comparison of TSR datasets.** For TableBank and SynthTabNet, we randomly sampled 1,000 images to assess color diversity and found that fewer than 10 were colored, indicating weak diversity. For TabRecSet and WTW, their diversity came from being captured by cameras in real world. **Annotation legend:** T: table bounding box (supports table detection); S: structural annotation (e.g., HTML tags, spanning); C: cell-level annotation (bounding box and content); H/X: final HTML/XML output or re-constructable from S and C; V: visual labels (e.g., simplicity, color, border style) provided in our dataset. **Definition of diversity:** A table image dataset is deemed as diverse if it supports at least two distinct and meaningful classification standards by which its images can be systematically categorized. These standards may include structural, visual attributes, allowing for multifaceted analysis and evaluation. For example, one classification standard could be based on structural complexity, such as whether a table contains spanning cells; another could be based on visual style, including attributes like border presence, background color, or watermark interference. A dataset that enables classification along both dimensions provides more comprehensive coverage for training and evaluating table structure recognition models.

Datasets	Diversity	Annotation	Domain	Collection methodology	Tables
TableNet (Ours)	True	S, C, H, V	Telecom (configurable)	See Figure 2	445K
PubTabNet	False	S, C, H	Medicine	rule-based collection	510K
FinTabNet	False	T, S, C, H	Finance	augmentation	113K
TableBank	True	T, H	General	rule-based collection	145K
SynthTabNet	True	T, S, C, H	General	augmentation	600K
ICDAR 2019	False	T, X	General	institutional contribution	3K
PubTables-1m	False	T, S, C, H	Medicine	rule-based collection	948K
TabRecSet	True	T, S, C, H	General	manual labelling	38K
WTW	True	S, C	General	manual labelling	14K
SciTSR	False	T, C	General	rule-based collection	15K

tions of questions or descriptions. Recent approaches like (Han et al. 2023; He et al. 2024; Xia et al. 2024; Yang et al. 2025) generate charts or multiple kinds of image data, but they rely on LLMs to generate HTML code, making the overall procedure uncontrollable and more likely to generate error code. In our work, we designed a multi-agent system that explicitly decomposes the table generation into schema planning, layout construction and content filling, ensuring controllable table image synthesis and TSR annotating.

Multi-Agent Table Generation System

LLMs demonstrate competitive performance in various tasks(Jian, Yang, and Jiao 2024; Jian et al. 2025), but passively predicting does not outperforms heuristic or simple agent tools on specific tasks like HTML structure generation. LLM-based multi-agent system is application that combine multiple LLMs with capabilities such as tool usage, planning, and memory. The LLM acts as the "brain" of the system interacting with users and execute key tasks. Planning helps decompose the request into manageable subtasks, which the system solves individually. Additionally, a reflection mechanism is integrated to refine execution. Tool usage enables the system to interact with the external and gather the necessary information to fulfill the user's request. Since the workflow of our system is clearly illustrated in Figure 1, therefore in this section, we describe how these capabilities are integrated into our table generation multi-agent, enabling configurable, domain-flexible, and semantically grounded table synthesis.

Workflow and Core Concept Embodiment

LLMs Our table generation system comprises three stage-specific agents coordinated by a core LLM. The core LLM interprets user requests—such as style, quantity, and domain—and orchestrates the workflow. A topic generation LLM produces table topics aligned with the specified or default telecommunication domain, ensuring semantic relevance. A header-infilling LLM and a body-infilling LLM then complete the HTML skeleton by inserting contents into the `<th>` and `<td>` tags. Our system is developed based on an open source codebase[1].

Planning The core LLM functions as a high-level planner that decomposes the task into structured subtasks with clear dependencies. Upon receiving a request, the Schema Agent determines table size and layout attributes, including row/column counts and spanning relations, and conditionally invokes a CSS generator for visual styling. It then generates a domain-relevant topic and constructs an initial HTML skeleton. Header and body infilling models are invoked sequentially after prerequisites are satisfied. Finally, the Filling Agent compares the filled and unfilled HTML to assess structural integrity and selectively regenerates the HTML if errors are detected. This multi-step, feedback-driven workflow ensures robust and controllable table synthesis.

Tool Using To support multi-step execution, the system integrates several tool modules: (1) a CSS style generator; (2) an HTML tags generator specifying size and spanning; (3) a structure validator that builds a matrix representation and

[1]https://github.com/WenmuZhou/TableGeneration/tree/main under MIT License.

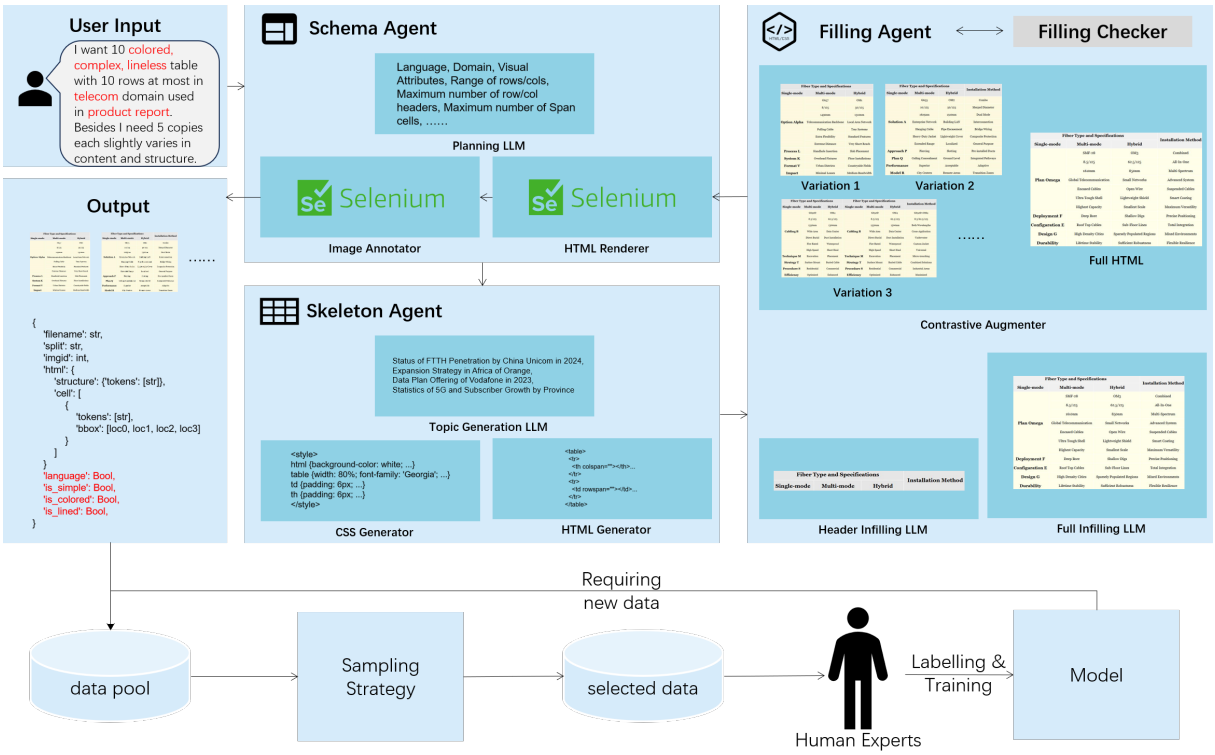

Figure 1: Workflow of our multi-agent system.

checks row equality; (4) a fallback HTML constructor for regenerating compliant tables; and (5) a Selenium tool for rendering table images and producing annotations.

Memory The system incorporates a two-level memory mechanism. Outer memory retains the multi-turn dialog history between the core LLM and the user, ensuring continuity and refinement. Inner memory tracks previously generated table topics to avoid redundancy and promote semantic diversity. Together, these memories enhance interaction quality and generation diversity.

Within the Filling Agent, we adopt a contrastive learning–inspired augmentation strategy (Le-Khac, Healy, and Smeaton 2020) to enhance the diversity and robustness of TableNet. We define four transformations—copy, delete, swap, and alter—applied at row/column or block levels. Copy inserts duplicates; delete removes elements only when structural validity is preserved; swap exchanges rows/-columns or blocks; alter adjusts row-level background colors. For each HTML structure, the body-infilling LLM generates multiple content variants (five without transformation and four with transformations), each assigned one operation. Row and column spanning regions are detected to prevent invalid modifications. This augmentation increases diversity and helps models learn fine-grained structural distinctions.

To validate generated tables, we designed a mixed strategy filling checker to rank structure correctness, topic relevance, and semantic consistency. Structural correctness is evaluated using a heuristic method that checks validity (Figure 3) and HTML tag misuse; topic relevance and semantic consistency are LLM-ranked. We verified that this filling checker can substitute for a well-trained human ranker.

After generation, we further use the data to finetune a TSR model under an active learning paradigm. This human-in-the-loop stage can be invoked at any time, since the data pool includes both labeled generated tables and unlabeled real-world tables, which are more valuable for TSR performance. We first apply a sampling strategy to select informative samples, then pass them to human experts for high-quality labeling and subsequent training.

Dataset Collection and Composition

Our dataset is constructed from three distinct sources: (1) agent-generated tables, (2) crawled tables from telecommunication-related PDF documents, and (3) crawled tables from Word documents, along with augmented open-source HTML table data. These three sources represent agent generation, manual labeling, and rule-based generation, respectively. The detailed dataset composition is available in Section .The pipelines and the final annotation format are illustrated in Figure 2. We also present examples of tables in Figure 4.

Agent-generating pipeline. We employed an 8-way parallel generation strategy, where each process corresponds to a unique combination of three binary attributes (simple, colored, lined). In six days, we generated about 75K Chinese and 34K English tables, now expanded to 445K. To enhance realism, we target structural, semantic, and visual fidelity. Structurally, we summarize eight types of complex

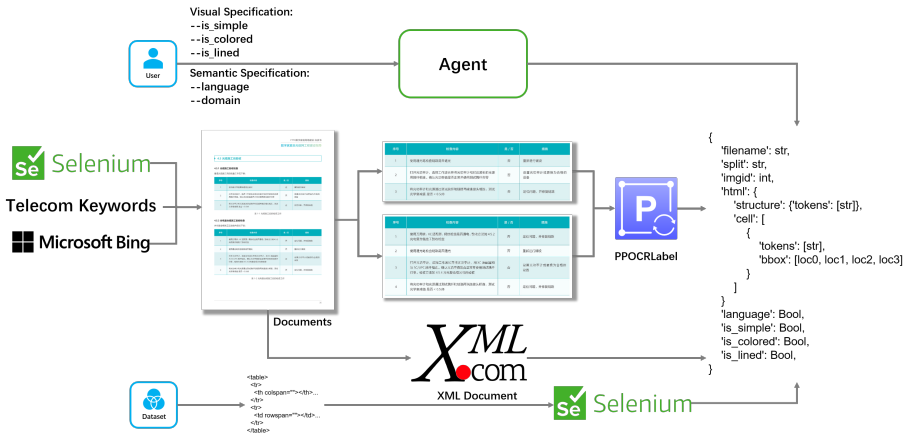

Figure 2: **Data collection pipeline.** From top to bottom: agent generating, web crawling and open source augmenting. **Annotation explanation:** is simple denotes whether the table contains cells spanning. is colored denotes whether the table includes any background color, colored borders, or non-black font colors. is lined specifies whether the cell borders are fully present; tables with only horizontal or vertical lines are considered not lined.

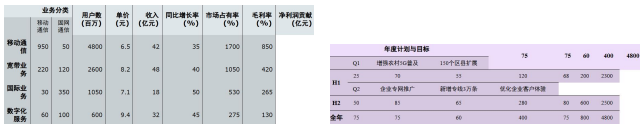

Figure 3: Invalid tables

span-free tables based on orientation, multi-level headers, and body-level spans, all supported by our system. Semantically, we capture scenario-dependent complexity (e.g., tech comparisons vs. financial reports), and our filling checker improves authenticity by detecting header–body mismatches and hallucinations. Visually, we regulate border thickness, line style, font and background colors via CSS heuristics, and incorporate secondary factors such as resolution variation and watermarking noise.

PDF and Word crawling & labeling pipeline. Given that China Telecom, China Mobile, China Unicom, and China Broadnet dominate the Chinese telecommunication industry, we used Selenium to construct queries combining their names with telecom-related keywords (e.g., FTTH, data plan;), restricting results to PDFs or Word files. For PDFs, due to the lack of reliable automatic extraction tools, we manually cropped and annotated 2.7K tables using PPOCRLabel over 30 days. For Word documents, we leveraged their zipped-XML structure to extract table markup, convert it to HTML, apply CSS via our generator, render images, and obtain annotations, yielding 600 tables despite the limited availability of Word files.

Open-source dataset augmentation pipeline. Since table understanding models typically take structured formats (e.g., HTML/Markdown) as input, we treated these structured inputs as outputs and rendered images and annotations using the same process as in the Word pipeline. Based on TABMWP (Lu et al. 2022), we generated 1K augmented tables. However, augmentation alone is constrained by data availability, limited controllability, and restricted variability in domain, structure, and visual style.

Experiments

Diversity and Filling Checker Analysis

Industry diversity groundness. To validate the capability of our multi-agent system to generate tables of multiple domains, we selected 16 industries of 8 sectors from Global Industry Classification Standard (GICS) and calculated their Spearman and Pearson correlation and Kendall's tau between our filling checker and well-trained human rankers. Besides, we repeated the procedure three times and calculated the standard deviation to ensure the stability of the results. According to (Zhang et al. 2019), our filling checker is capable of substituting well-trained human rankers and thus proves the semantic consistency and the multiple domain table generation ability. The detailed results are available as supplementary materials.

Across most sectors, the correlation metrics remain consistently high (mostly excels 0.8). Besides, we also evaluated average iterations for our system to correct a table and corresponding metrics after disturbing on structure, topic, and semantics in Table 2. These indicate 1) strong semantic alignment within each language group and 2) filling checker is capable of ranking in replace of well-trained human rankers (Zhang et al. 2019). This experiment provides a fine-grained view of the semantic cohesion within industrial table data and offers empirical grounding for further bilingual alignment or domain-adaptive table generation research.

Style diversity groundness. By observeing our crawled real-world scenario tables, we described some key factor that determines the general veracity of a table. A distribution of configurable styles of TableNet is shown in Figure 5. As described in Section , our system is capable of generating multiple kinds of tables. For detailed table diversity, we evaluated detailed distribution of line style, structure complexity of our TableNet. Figure shown in supplementary materials

Technology	Performance Metrics				
	Speed (Mbps)	Latency (ms)	Bandwidth (GHz)	Range (km)	Reliability (%)
5G NR	1000	10	10	2	99.99
DSL	100	50	1	5	99.5
Fiber Optic	5000	5	20	80	99.9
Wi-Fi 6	3000	20	8	1	99.7
Satellite	100	500	0.5	35000	98.5
LTE	150	30	5	10	99.6
Li-Fi	7000	3	15	10	99.8
Bluetooth 5	50	50	2	0.1	95

(a) complex colored lined vertical table

中国八大电信运营商网络指标比一览							
运营商	最高理论带宽（单位：）	每条用户线最大带宽（单位：）	每条用户线平均带宽	网络覆盖范围	可选增值服务	业务入网资费标准
中国移动	120	30	500	138	智能家庭管理、视频彩铃业务	全国APP自助办理，营业厅开业
中国联通	120	40	600	88	5G优先接入	网络线上自助办理，线下营业厅
中国电信	120	30	680	138	超高速5G网络覆盖	大屏会员、智能语音管理
广电网络	90	20	300	50	广电视频专属业务	网络智能WiFi设置
中国铁塔	-	-	-	-	无直接销售	无直接销售
虚拟运营商（以移动）	70	10	200	38	依靠合作运营商提供服务	出国网络设备服务
备注	每费用结果评估	在普通家宅之	以公共事业物业网络为例	即在增强型设备情况下	具体情况以当地网络覆盖为准	增强型设备要每年更换设备

(b) complex zebra lineless matrix table

Region	Technology Type	Infrastructure Status	Service Provider
North America	Fiber Optic	Operational	Provider A
Asia-Pacific	DSL	Under Development	Provider B
Europe	Cable Broadband	Maintenance	Provider C
Africa	Satellite	Planned	Provider D
South America	FTTH	Operational	Provider E
Middle East	LTE	Under Testing	Provider F
Australia	ADSL	Deprecated	Provider G
India	GPON	Expansion Phase	Provider H
China	Hybrid Fiber-Coaxial	Stable	Provider I
Japan	Wireless	Emerging	Provider J

(c) simple colored lined vertical table

运营商名称	中国移动	中国电信	中国电信	日本AT&T	英国Vodafone	德国Deutsche Telecom	日本NTT Docomo	韩国SK Telecom	印度Bharti Airtel
运营商简称	大学	大学	大学	大学	大学	大学	中学	中学	大学
网络覆盖率(%)	95.3	92.3	94.8	86.9	98.7	91.2	86.5	87.3	76.4
客户服务评分(1-5)	4.3	4.3	4.6	4.0	4.1	4.2	3.9	4.1	3.6
千兆网速(Mbps)	48.5	46.2	58.1	42.7	38.9	40.5	36.8	43.2	26.6
用户满意度评分(1-5)	4.4	4.2	4.3	4.0	3.9	4.1	4.0	4.2	3.3

(d) simple zebra lined horizontal table

Figure 4: examples in TableNet

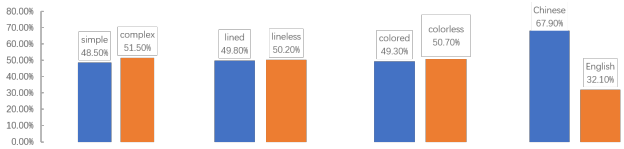

Figure 5: Generated data composition

Table 2: Correlation between human and filling checker ranks. Struct. and Sem. represent structure and semantics.

Complexity	Dim.	Avg Iters	Spear.	Pear.	Ken.
Simple	Struct.	1.03	0.8203	0.8111	0.7928
	Topic	1.07	0.8004	0.8085	0.6990
	Sem.	1.03	0.9253	0.8914	0.8522
Complex	Struct.	1.11	0.8826	0.8751	0.8832
	Topic	1.08	0.7352	0.7408	0.6457
	Sem.	1.17	0.8753	0.8411	0.8019

indicates our TableNet covers most of given classifications where existing TSR datasets fail to achieve due to the limitations caused by collection methodology.

Experiment with LLMs

We used TEDS (tree edit distance-based simlarity)(Nassar et al. 2022) as metric. To evaluate the effectiveness of our TableNet, we conducted a fine-tuning experiment using Qwen2-VL-2B on our TableNet Chinese train split, which contains about 48K tables. The full-parameter training process was conducted using 2 NVIDIA 4090 RTX GPUs with a batch size of 1 and a learning rate of 1e-4. The model was trained for 2 epochs, and the entire training process took approximately 8 hours. Training, testing split and corresponding subset for our training process will also be available. For baselines, we selected large-scale multimodal LLMs as baselines: Qwen2-VL-72B(Wang et al. 2024), GPT(Achiam et al. 2023), Claude(Anthropic 2024) and Grok(xAI 2024) series. To avoid possible negative impact caused by inaccurate prompt design, we applied 1-shot and 5-shot in-context learning to Qwen2-VL-72B by integrating some training samples.

Table 4 summarizes the results. Models trained on TableNet show strong and stable performance across diverse table styles, whereas all baselines exhibit clear performance gaps between simple vs. complex structures, color vs. colorless layouts, and lined vs. lineless designs. This confirms that table diversity poses substantial generalization challenges and highlights the value of controllable generation and detailed annotations.Structural complexity—particularly row/column spans—remains the most difficult factor, as it requires precise spatial and hierarchical reasoning. Our fine-tuned Qwen2-VL-2B(FT) markedly narrows this gap and outperforms larger models on complex structures. Color variations also influence performance: while colored tables introduce richer context but higher variability, Qwen2-VL-2B(FT) remains robust (0.874 vs. 0.892), whereas zero-/few-shot large LLMs fluctuate more. Line presence further tests models' ability to infer implicit structure; again, our model maintains consistent accuracy.Overall, even state-of-the-art LLMs struggle without targeted supervision, underscoring the necessity of structurally grounded training for TSR.

Table 3: TSR performance on unseen real-world data of models trained on different existing datasets.

dataset	TEDS
TableNet(Ours)	**0.7403**
PubTabNet	0.5041
FinTabNet	0.4495
SynthTabNet	0.5242
TableBank	0.5401

To further explained the effectiveness compared to other existing datasets, we used the crawled real-world tables as test set, and fine-tuned Qwen2-VL-2B on existing datasets using the same number of samples and identical training settings, and separately fine-tuned it on TableNet with crawled data excluded. We then evaluated models on unseen real-world tables spanning diverse structural, semantic, and visual styles. The model trained on TableNet achieved a TEDS of 0.7403, substantially outperforming the other-trained model, demonstrating the superior real-world generalizability brought by TableNet. The results are shown in Table 3.

Ablation Study

LLM-driven data generation often suffers from structural instability, prompting us to evaluate LLMs' end-to-end table synthesis capability using structure-only TEDS (Zhong, ShafieiBavani, and Jimeno Yepes 2020). For simple tables,

Table 4: TEDS of HTML on TableNet and subsets.

Models	All	is simple		is colored		is lined	
		Simple	Complex	Colored	Colorless	Lined	Lineless
Qwen							
Qwen2-VL-2B(FT)	**0.877**	0.860	**0.912**	**0.874**	**0.892**	**0.891**	**0.861**
Qwen2-VL-72B	0.721	0.822	0.604	0.726	0.713	0.711	0.729
Qwen2-VL-72B(1-shot)	0.696	0.834	0.587	0.706	0.685	0.699	0.695
Qwen2-VL-72B(5-shot)	0.688	0.791	0.576	0.695	0.682	0.689	0.687
GPT							
GPT-4.5-preview	0.614	0.872	0.494	0.732	0.501	0.591	0.634
GPT-4o-all	0.672	0.794	0.487	0.690	0.655	0.664	0.687
Claude							
Claude-sonnet-4	0.620	0.904	0.517	0.687	0.533	0.621	0.618
Claude-sonnet-4-thinking	0.706	0.917	0.561	0.693	0.719	0.749	0.648
Claude-opus-4	0.691	**0.921**	0.523	0.704	0.675	0.722	0.661
Grok							
Grok-4	0.697	0.892	0.538	0.731	0.642	0.678	0.707

Table 5: Table Fidelity using Structure-only TEDS of HTML generated by LLM through directly prompting. Small texts after ones means TEDS structure infidelity introduced by LLM infilling, which is lowest.

Methods	Simple		Complex	
	with	without	with	without
Qwen2.5	0.954	0.978	0.782	0.835
DeepSeek-R1	0.938	0.974	0.940	0.901
GPT-4o	0.980	0.978	0.931	0.928
Agent Tool	1 -0.004	1 -0.003	1 -0.012	1 -0.009

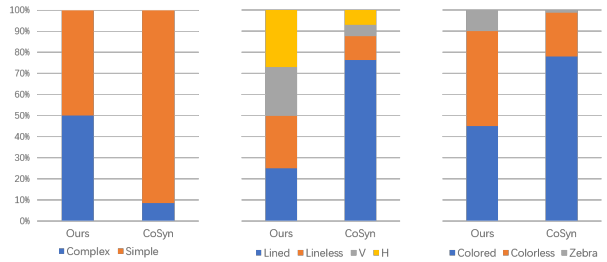

Figure 6: Compared with existing CoSyn pipeline, which is an LLM end-to-end method. In the middle subfigure, H represents horizontally lined and V represents vertically lined.

we instructed the LLM to produce HTML with specified row/column counts; for complex tables, we supplied explicit row- and column-span matrices that are difficult to express in natural language. As shown in Table 5, LLMs frequently fail to reproduce accurate structures—even in simple cases—while rule-based HTML generation reliably yields correct layouts and avoids structural infidelity through controlled prompting. These results demonstrate that, compared with directly generating tables via LLMs, an agent that invokes LLMs only at appropriate stages provides significantly more stable and accurate structure synthesis.

To assess whether agent-based design improves dataset quality, we evaluated the recent CoSyn pipeline for table-image and QA generation (Yang et al. 2025). Using it to generate 1,000 tables, only 85 contained complex structures such as merged rows/columns; most followed a simple stacked `<tr><td>` format. We also evaluate the color and line style diversity, results are shown in Figure 6.

After strengthening its HTML prompt to enforce structural complexity, the enhanced baseline achieved an average structural score of 4.31 (out of 5) under our checker. In contrast, our agent-based pipeline reached 4.79 on average and produced structurally correct tables with only 1.11 fallback iterations per sample.

Experiment on Active Learning TSR

The active learning procedure is detailed in the following. We use an multi-agent system to generate new data instances and corresponding labels, not just selecting from the unlabeled set $\mathcal{U}$. The LLM serves as a powerful tool throughout the AL loop, starting by annotating an initial dataset ($\mathcal{L}_{\text{init}}$) to address the cold-start problem. In the Query step, we selects informative instances from $\mathcal{U}$. These instances are then annotated in the Annotation step, either automatically or manually. The newly labeled data is added to the labeled set $\mathcal{L}$, and the target model f_θ is updated. The process iterates until the given annotation budget is met. This method overcomes challenges in traditional AL by using LLMs for data generation and automatic agent tools for annotation, reducing reliance on human annotators.

Algorithm 1: Active Learning

Input: Unlabeled dataset $\mathcal{U}$, Model M, Budget k
Output: Trained model f_θ, Labeled dataset $\mathcal{L}$
$\mathcal{L}_{\text{init}}, \mathcal{U} \leftarrow \text{Initialize}$
$f_\theta \leftarrow \text{Train}(\mathcal{L}_{\text{init}})$
while not $\text{Stop}(k, f_\theta, M)$ **do**
  $x \leftarrow \text{Query}(f_\theta, \mathcal{U}, M)$
  $(x, y) \leftarrow \text{Annotate}(x, M)$
  if $x \in \mathcal{U}$ **then** $\mathcal{U} \leftarrow \mathcal{U} \setminus \{x\}$
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{(x, y)\}$
  $f_\theta \leftarrow \text{Train}(\mathcal{L})$
end while
return $f_\theta^*, \mathcal{L}$

Algorithm 2: k-Center-Greedy

Input: Data points $\{x_i\}_{i=1}^n$, initial set s_0, budget b
Output: s
$s \leftarrow s_0$
repeat
  $u \leftarrow \arg\max_{i \in [n] \setminus s} \min_{j \in s} \Delta(x_i, x_j)$
  $s \leftarrow s \cup \{u\}$
until $|s| = b + |s_0|$
return $s \setminus s_0$

Based on Qwen2-VL-2B(FT) (hereafter the base model), we conducted an active-learning fine-tuning study using a diversity-based strategy because TableNet exhibits diversity in row/column counts, merged-cell locations, colors, etc. Training used 2 × NVIDIA RTX 4090 GPUs with LoRA fine-tuning for 1 epoch; TEDS served as the evaluation metric. We adopted CoreSet (Sener and Savarese 2017), a greedy k-center shown in Algorithm 2 where we try to choose b centers such that the largest distance between a data point and its nearest center is minimized, formally trying to solve shown below, where s_0 denotes the unlabeled dataset, b is the annotation budget, and s_1 represents the set of samples selected for annotation.

$$\min_{s_1 : |s_1| = b} \max_{i} \min_{j \in s_1 \cup s_0} \Delta(x_i, x_j). \tag{1}$$

We use it as a querying strategy shown in Algorithm 1, and compared it to random sampling (RSB), hard-example mining (HE) (Liu et al. 2025; Shrivastava, Gupta, and Girshick 2016), and perplexity-based uncertainty sampling (PPL). We extracted the patched vision embeddings $e \in \mathbb{R}^{patch \times dim}$ from the vision tower of base model and concatenated its max- and mean-pooled embedding into a vector $c \in \mathbb{R}^{2 \times dim}$ to perform the method.

As shown in Figure 7, diversity-based CoreSet method demonstrates stably higher performance that baselines, including random sampling, perplexity-based uncertainty sampling, and hard examples mining. Besides, We can also observe that active learning method achieved a comparable performance compared to baselines while reducing the number of sample by at least 50% in both low-data regime and standard-data regime by comparing the number of training samples of the same performance. For example, model fine-tuned on 10k actively selected samples achieved a TEDS performance around 0.973 while other baselines used more than 20k, even 40k training samples.

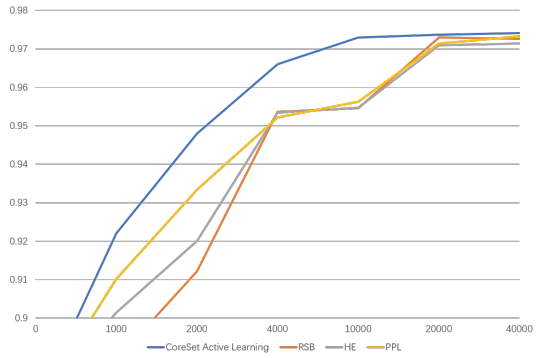

Figure 7: TSR Experiment results

Limitations

Although our system supports configurable parameters, it is still bounded by the pretraining distribution and reasoning capabilities of the underlying LLM. As a result, the diversity of generated content may not fully capture edge cases or highly domain-specific table formats. Furthermore, the quality of generated table images may vary due to model's knowledge of user specified domain and language, leading to irrelevant table contents or inconsistencies within semantic contents.

Conclusion

In this work, we introduce TableNet, a new dataset for TSR mainly sourced from autonomous generation. Complementing the dataset, we develop the first autonomous table generation and recognition multi-agent system that facilitates TSR and supports controllable table synthesis and automatic annotation through user-defined parameters. It enables scalable, flexible, and research-oriented data construction, and demonstrates high efficiency with the capacity to generate large volumes of structurally reliable tables. The scalability of our collection process, together with empirical results, highlights the significance of both TableNet and the system as TSR tools for TSR. We believe that integrating a powerful generation agent with TableNet will meaningfully push TSR.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ajayi, K.; Zhang, L.; He, Y.; and Wu, J. 2024. Uncertainty Quantification in Table Structure Recognition. In *2024 IEEE International Conference on Information Reuse and Integration for Data Science (IRI)*, 1–6. IEEE.

Anand, A.; Jaiswal, R.; Bhuyan, P.; Gupta, M.; Bangar, S.; Imam, M. M.; Shah, R. R.; and Satoh, S. 2023. Tc-ocr: Tablecraft ocr

for efficient detection & recognition of table structure & content. In *Proceedings of the 1st International Workshop on Deep Multimodal Learning for Information Retrieval*, 11–18.

Anthropic. 2024. Claude 3 Model Family System Card.

Beluch, W. H.; Genewein, T.; Nürnberger, A.; and Köhler, J. M. 2018. The power of ensembles for active learning in image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9368–9377.

Carbune, V.; Mansoor, H.; Liu, F.; Aralikatte, R.; Baechler, G.; Chen, J.; and Sharma, A. 2024. Chart-based reasoning: Transferring capabilities from llms to vlms. *arXiv preprint arXiv:2403.12596*.

Cascante-Bonilla, P.; Wu, H.; Wang, L.; Feris, R. S.; and Ordonez, V. 2022. Simvqa: Exploring simulated environments for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5056–5066.

Cesarini, F.; Marinai, S.; Sarti, L.; and Soda, G. 2002. Trainable table location in document images. In *2002 International Conference on Pattern Recognition*, volume 3, 236–240. IEEE.

Chi, Z.; Huang, H.; Xu, H.-D.; Yu, H.; Yin, W.; and Mao, X.-L. 2019. Complicated table structure recognition. *arXiv preprint arXiv:1908.04729*.

Fang, J.; Tao, X.; Tang, Z.; Qiu, R.; and Liu, Y. 2012. Dataset, ground-truth and performance metrics for table detection evaluation. In *2012 10th IAPR International Workshop on Document Analysis Systems*, 445–449. IEEE.

Gal, Y.; Islam, R.; and Ghahramani, Z. 2017. Deep bayesian active learning with image data. In *International conference on machine learning*, 1183–1192. PMLR.

Gao, L.; Huang, Y.; Déjean, H.; Meunier, J.-L.; Yan, Q.; Fang, Y.; Kleber, F.; and Lang, E. 2019. ICDAR 2019 competition on table detection and recognition (cTDaR). In *2019 International conference on document analysis and recognition (ICDAR)*, 1510–1515. IEEE.

Gao, L.; Yi, X.; Jiang, Z.; Hao, L.; and Tang, Z. 2017. ICDAR2017 competition on page object detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, 1417–1422. IEEE.

Gissin, D.; and Shalev-Shwartz, S. 2019. Discriminative active learning. *arXiv preprint arXiv:1907.06347*.

Göbel, M.; Hassan, T.; Oro, E.; and Orsi, G. 2013. ICDAR 2013 table competition. In *2013 12th international conference on document analysis and recognition*, 1449–1453. IEEE.

Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; and Babenko, A. 2021. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34: 18932–18943.

Han, Y.; Zhang, C.; Chen, X.; Yang, X.; Wang, Z.; Yu, G.; Fu, B.; and Zhang, H. 2023. Chartllama: A multimodal llm for chart understanding and generation. *arXiv preprint arXiv:2311.16483*.

Hashmi, K. A.; Liwicki, M.; Stricker, D.; Afzal, M. A.; Afzal, M. A.; and Afzal, M. Z. 2021. Current status and performance analysis of table recognition in document images with deep neural networks. *IEEE Access*, 9: 87663–87685.

He, W.; Xi, Z.; Zhao, W.; Fan, X.; Ding, Y.; Shan, Z.; Gui, T.; Zhang, Q.; and Huang, X. 2024. Distill Visual Chart Reasoning Ability from LLMs to MLLMs. *arXiv preprint arXiv:2410.18798*.

Itonori, K. 1993. Table structure recognition based on textblock arrangement and ruled line position. In *Proceedings of 2nd International Conference on Document Analysis and Recognition (ICDAR'93)*, 765–768. IEEE.

Jian, C.; Yang, K.; and Jiao, Y. 2024. Tri-Level Navigator: LLM-Empowered Tri-Level Learning for Time Series OOD Generalization. *Advances in Neural Information Processing Systems*, 37: 110613–110642.

Jian, C.; Yang, K.; Ouyang, Y.; and Ye, X. 2025. Stable Preference Optimization for LLMs: A Bilevel Approach Beyond Direct Preference Optimization. *arXiv preprint arXiv:2507.07723*.

Johnson-Roberson, M.; Barto, C.; Mehta, R.; Sridhar, S. N.; Rosaen, K.; and Vasudevan, R. 2016. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*.

Kafle, K.; Price, B.; Cohen, S.; and Kanan, C. 2018. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5648–5656.

Kahou, S. E.; Michalski, V.; Atkinson, A.; Kádár, Á.; Trischler, A.; and Bengio, Y. 2017. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*.

Kayal, P.; Anand, M.; Desai, H.; and Singh, M. 2021. ICDAR 2021 competition on scientific table image recognition to LaTeX. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part IV 16*, 754–766. Springer.

Kieninger, T.; and Dengel, A. 1999. The t-recs table recognition and analysis system. In *Document Analysis Systems: Theory and Practice: Third IAPR Workshop, DAS'98 Nagano, Japan, November 4–6, 1998 Selected Papers 3*, 255–270. Springer.

Koci, E.; Thiele, M.; Lehner, W.; and Romero, O. 2018. Table recognition in spreadsheets via a graph representation. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, 139–144. IEEE.

Koci, E.; Thiele, M.; Romero, O.; and Lehner, W. 2017. Table identification and reconstruction in spreadsheets. In *Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings 29*, 527–541. Springer.

Koci, E.; Thiele, M.; Romero, O.; and Lehner, W. 2019. A genetic-based search for adaptive table recognition in spreadsheets. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 1274–1279. IEEE.

Le-Khac, P. H.; Healy, G.; and Smeaton, A. F. 2020. Contrastive representation learning: A framework and review. *Ieee Access*, 8: 193907–193934.

Li, K.; Wigington, C.; Tensmeyer, C.; Zhao, H.; Barmpalios, N.; Morariu, V. I.; Manjunatha, V.; Sun, T.; and Fu, Y. 2020a. Cross-domain document object detection: Benchmark suite and method. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12915–12924.

Li, M.; Cui, L.; Huang, S.; Wei, F.; Zhou, M.; and Li, Z. 2020b. Tablebank: Table benchmark for image-based table detection and recognition. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 1918–1925.

Li, P.; He, Y.; Yashar, D.; Cui, W.; Ge, S.; Zhang, H.; Fainman, D. R.; Zhang, D.; and Chaudhuri, S. 2023. Table-gpt: Table-tuned gpt for diverse table tasks. *arXiv preprint arXiv:2310.09263*.

Li, S.; and Tajbakhsh, N. 2023. Scigraphqa: A large-scale synthetic multi-turn question-answering dataset for scientific graphs. *arXiv preprint arXiv:2308.03349*.

Liu, H.; Li, X.; Gong, M.; Liu, B.; Wu, Y.; Jiang, D.; Liu, Y.; and Sun, X. 2024. Grab what you need: Rethinking complex table structure recognition with flexible components deliberation. In

Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 3603–3611.

Liu, L.; Liang, Y.; Yan, X.; Huangfu, L.; Samtani, S.; Yu, Z.; Zhang, Y.; and Zeng, D. D. 2025. Hard Sample Mining: A New Paradigm of Efficient and Robust Model Training. *IEEE Transactions on Neural Networks and Learning Systems*.

Long, R.; Wang, W.; Xue, N.; Gao, F.; Yang, Z.; Wang, Y.; and Xia, G.-S. 2021. Parsing table structures in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 944–952.

Lu, P.; Qiu, L.; Chang, K.-W.; Wu, Y. N.; Zhu, S.-C.; Rajpurohit, T.; Clark, P.; and Kalyan, A. 2022. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.

Nassar, A.; Livathinos, N.; Lysak, M.; and Staar, P. 2022. Tableformer: Table structure understanding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4614–4623.

Pang, C.; Cao, Y.; Yang, C.; and Luo, P. 2024. Uncovering limitations of large language models in information seeking from tables. *arXiv preprint arXiv:2406.04113*.

Pyreddy, P.; and Croft, W. B. 1997. Tintin: A system for retrieval in text tables. In *Proceedings of the second ACM international conference on Digital libraries*, 193–200.

Qasim, S. R.; Mahmood, H.; and Shafait, F. 2019. Rethinking table recognition using graph neural networks. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 142–147. IEEE.

Raja, S.; Mondal, A.; and Jawahar, C. 2020. Table structure recognition using top-down and bottom-up cues. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, 70–86. Springer.

Rus, D.; and Subramanian, D. 1997. Customizing information capture and access. *ACM Transactions on Information Systems (TOIS)*, 15(1): 67–101.

Schreiber, S.; Agne, S.; Wolf, I.; Dengel, A.; and Ahmed, S. 2017. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, 1162–1167. IEEE.

Sener, O.; and Savarese, S. 2017. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*.

Shigarov, A.; Mikhailov, A.; and Altaev, A. 2016. Configurable table structure recognition in untagged pdf documents. In *Proceedings of the 2016 ACM symposium on document engineering*, 119–122.

Shrivastava, A.; Gupta, A.; and Girshick, R. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 761–769.

Smock, B.; Pesala, R.; and Abraham, R. 2022. PubTables-1M: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4634–4642.

Somvanshi, S.; Das, S.; Javed, S. A.; Antariksa, G.; and Hossain, A. 2024. A survey on deep tabular learning. *arXiv preprint arXiv:2410.12034*.

Su, A.; Wang, A.; Ye, C.; Zhou, C.; Zhang, G.; Chen, G.; Zhu, G.; Wang, H.; Xu, H.; Chen, H.; et al. 2024. Tablegpt2: A large multimodal model with tabular data integration. *arXiv preprint arXiv:2411.02059*.

Sui, Y.; Zhou, M.; Zhou, M.; Han, S.; and Zhang, D. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, 645–654.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

xAI. 2024. Grok-1 Model Card.

Xia, R.; Zhang, B.; Ye, H.; Yan, X.; Liu, Q.; Zhou, H.; Chen, Z.; Ye, P.; Dou, M.; Shi, B.; et al. 2024. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185*.

Xia, Y.; Mukherjee, S.; Xie, Z.; Wu, J.; Li, X.; Aponte, R.; Lyu, H.; Barrow, J.; Chen, H.; Dernoncourt, F.; et al. 2025. From selection to generation: A survey of llm-based active learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 14552–14569.

Xue, W.; Li, Q.; and Tao, D. 2019. Res2tim: Reconstruct syntactic structures from table images. In *2019 international conference on document analysis and recognition (ICDAR)*, 749–755. IEEE.

Yang, F.; Hu, L.; Liu, X.; Huang, S.; and Gu, Z. 2023. A large-scale dataset for end-to-end table recognition in the wild. *Scientific Data*, 10(1): 110.

Yang, K.; Ren, J.; Zhu, Y.; and Zhang, W. 2018. Active learning for wireless IoT intrusion detection. *IEEE Wireless Communications*, 25(6): 19–25.

Yang, Y.; Patel, A.; Deitke, M.; Gupta, T.; Weihs, L.; Head, A.; Yatskar, M.; Callison-Burch, C.; Krishna, R.; Kembhavi, A.; and Clark, C. 2025. Scaling Text-Rich Image Understanding via Code-Guided Synthetic Multimodal Data Generation. In Che, W.; Nabende, J.; Shutova, E.; and Pilehvar, M. T., eds., *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 17486–17505. Vienna, Austria: Association for Computational Linguistics. ISBN 979-8-89176-251-0.

Yoo, D.; and Kweon, I. S. 2019. Learning loss for active learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 93–102.

Zha, L.; Zhou, J.; Li, L.; Wang, R.; Huang, Q.; Yang, S.; Yuan, J.; Su, C.; Li, X.; Su, A.; et al. 2023. Tablegpt: Towards unifying tables, nature language and commands into one gpt. *arXiv preprint arXiv:2307.08674*.

Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Zheng, M.; Feng, X.; Si, Q.; She, Q.; Lin, Z.; Jiang, W.; and Wang, W. 2024. Multimodal table understanding. *arXiv preprint arXiv:2406.08100*.

Zheng, X.; Burdick, D.; Popa, L.; Zhong, X.; and Wang, N. X. R. 2021. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 697–706.

Zhong, X.; ShafieiBavani, E.; and Jimeno Yepes, A. 2020. Image-based table recognition: data, model, and evaluation. In *European conference on computer vision*, 564–580. Springer.

Zhong, X.; Tang, J.; and Yepes, A. J. 2019. Publaynet: largest dataset ever for document layout analysis. In *2019 International conference on document analysis and recognition (ICDAR)*, 1015–1022. IEEE.

Zhou, Y.; Cheng, M.; Mao, Q.; Liu, Q.; Xu, F.; Li, X.; and Chen, E. 2024. Enhancing Table Recognition with Vision LLMs: A Benchmark and Neighbor-Guided Toolchain Reasoner. *arXiv preprint arXiv:2412.20662*.

Zhu, Y.; and Yang, K. 2019. Tripartite active learning for interactive anomaly discovery. *IEEE Access*, 7: 63195–63203.

Supplementary Experiment Results

Experiments on Multiple Industries

To evaluate the semantic consistency of tables across different Global Industry Classification Standard (GICS) sectors and industries, we conducted a comprehensive correlation analysis between Chinese and English tables. Specifically, we computed the average pairwise semantic correlation between corresponding tables in the two languages under each industry category. The metrics include Spearman's rank correlation (Spear.), Pearson correlation (Pear.), and Kendall's Tau (Ken.), each reported as mean ± half standard deviation to reflect statistical stability. Results are shown in Table 6

Each industry is evaluated separately, with two rows per industry: the first row corresponds to the semantic correlations for Chinese tables, and the second row to their English counterparts.

Detailed Diversity Analysis

Based on the obeservation of crawled real-world tables, we divided tables into 5 types based on line type and 6 types based on table headers. As shown in Figure 8, our system is capable of generating given types of tables. In Figure 8, fully lined means every cell has complete borderlines, and horizontally/vertically lineless mean cells miss their horizontal/vertical border lines. Lined Headers means only header cells has borderlines. Matrix/verticle/horizontal tables means they have both/top headers/left headers.

However, existing datasets always fail to cover these types. For example, PubTabNet and FinTabNet is collected from scientific articles and is mostly consisted of verticle single header tables that only contain horizontal lines with black and white color. TableBank is mostly consisted of fully lined verticle single header tables which harms the ability of models to recognize lineless tables.

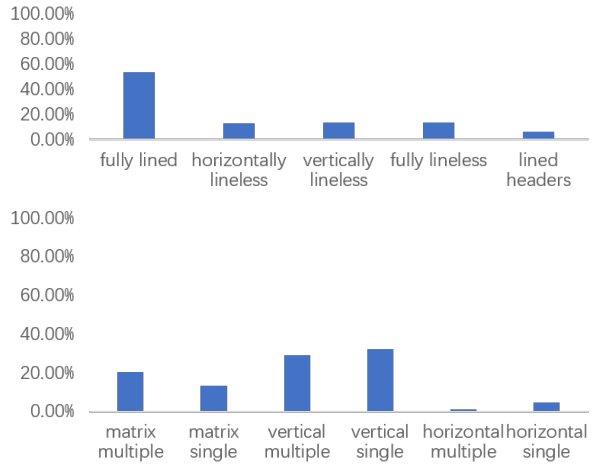

Figure 8: Detailed TableNet composition

Document Crawling Keywords

We crawled Microsoft search engine using Selenium by querying in format:

```
company abbreviation + telecom-related
      keyword + filetype:pdf/doc
```

Keywords are shown in Table 7.

LLM HTML Generation Prompts

Structure-Only Generation Prompt

Please generate an HTML table with {no_of_rows} rows and {no_of_cols} columns containing {content}. Only the `<tr>`, `<td>`, `</tr>`, and `</td>` tags should be used in the sequence.

Please return the generated table in the following JSON format: {"html": "HTML table string"}

The table should include cells that span multiple rows or columns. Below are two matrices indicating the row spans and column spans: Row span matrix: {row_spans_matrix} Column span matrix: {col_span_matrix}

Topic Generation Prompt

Please give me {copy} highly domain-relevant {lang} phrases related to the field of {domain}. Each phrase should be as detailed and specific as possible. Then return the generated phrases in the following JSON format: {"phrase": [phrase1, phrase2, phrase3, phrase4, phrase5]} The following topics have already been used. Please do not repeat these topics or generate content that is highly similar to them: [Used topics]

HTML Header Fulfilling Prompt

Below is an HTML table in the domain of {domain}, with the topic: {topic}, and the table language is {lang}. The HTML sequence to be filled is as follows: {HTML_CODE}

The structure of the HTML table is complete – the header section (<th> tags) already contains the column names, while the body section (<td> tags) is empty. Without making any changes to the original HTML structure, attributes, or tags, please fill in the content of the table body (<td> tags) based on the given topic and existing headers. Ensure that each cell in the table contains distinct and meaningful content.

Generate {copy} different versions of the filled HTML table. Then return the results in the following JSON format: `{"html": [list of filled HTML tables]}`

LLM Experiment and Few-Shot Prompt

Read the table content from the image and convert it into an HTML table format. The HTML must start with <html><body><table> and end with </table></body></html> and the entire HTML should be output in a single line. Your output must contain only the HTML that begins with <html><body><table> and ends with </table></body></html> and nothing else.

Examples: HTML1 HTML2 ...

HTML Body Fulfilling Prompt

You will act as a strict HTML editor. You are tasked with processing a table in the {domain} domain, with the topic: {topic}, and the table language is {lang}. The HTML snippet to be completed is as follows: {HTML_CODE}

The above is the original HTML paragraph for the table header. Without making any changes to the original HTML structure, please fill in the missing column names inside each <th> tag.

Important rules:

Table 6: Average semantic correlation adding or subtracting half standard deviation of multiple Global Industry Classification Standard (GICS) industries' tables. For each industry, the upper results line represents Chinese tables and the other line represents English tables.

Sector	Industry	Spear.	Pear.	Ken.
Energy	Equipment & Service	0.8021 ± 0.015 0.8204 ± 0.016	0.7954 ± 0.007 0.8080 ± 0.015	0.7311 ± 0.007 0.7551 ± 0.010
	Consumable Fuel	0.8344 ± 0.021 0.8121 ± 0.002	0.8047 ± 0.010 0.7853 ± 0.013	0.7612 ± 0.005 0.7224 ± 0.018
Material	Chemicals	0.8006 ± 0.017 0.8788 ± 0.027	0.8105 ± 0.015 0.8446 ± 0.017	0.7290 ± 0.022 0.8054 ± 0.020
	Metals & Mining	0.8951 ± 0.027 0.9091 ± 0.006	0.8848 ± 0.028 0.9004 ± 0.005	0.8109 ± 0.019 0.7923 ± 0.016
Health Care	Biotechnology	0.8516 ± 0.016 0.8607 ± 0.016	0.8476 ± 0.015 0.8394 ± 0.017	0.7577 ± 0.015 0.7509 ± 0.004
	Pharmaceuticals	0.8585 ± 0.007 0.8433 ± 0.015	0.8347 ± 0.010 0.8172 ± 0.011	0.7544 ± 0.008 0.7712 ± 0.017
Financials	Banks	0.8821 ± 0.018 0.9007 ± 0.020	0.8543 ± 0.019 0.8687 ± 0.022	0.7654 ± 0.009 0.8147 ± 0.027
	Insurance	0.7963 ± 0.014 0.8191 ± 0.011	0.7896 ± 0.009 0.8000 ± 0.004	0.7089 ± 0.002 0.6891 ± 0.008
Information Technology	Communication Equipment	0.8598 ± 0.016 0.8437 ± 0.022	0.8239 ± 0.011 0.8171 ± 0.011	0.7653 ± 0.016 0.7096 ± 0.001
	Semiconductor Equipment	0.8243 ± 0.015 0.8457 ± 0.012	0.8145 ± 0.015 0.8019 ± 0.016	0.7505 ± 0.010 0.7174 ± 0.004
Communication Services	Telecommunications	0.8826 ± 0.006 0.8897 ± 0.009	0.8224 ± 0.007 0.8656 ± 0.010	0.7338 ± 0.004 0.7560 ± 0.005
	Media	0.8537 ± 0.010 0.8751 ± 0.011	0.8348 ± 0.010 0.8322 ± 0.010	0.7299 ± 0.004 0.7402 ± 0.008
Utilities	Electric Utilities	0.8941 ± 0.009 0.8656 ± 0.013	0.8781 ± 0.06 0.8346 ± 0.015	0.7885 ± 0.002 0.7750 ± 0.006
	Gas Utilities	0.8338 ± 0.012 0.8052 ± 0.013	0.7953 ± 0.011 0.7940 ± 0.012	0.7287 ± 0.006 0.6800 ± 0.008
Real Estate	Equity Real Estate Trusts	0.8208 ± 0.015 0.8236 ± 0.014	0.7890 ± 0.015 0.8163 ± 0.005	0.6868 ± 0.006 0.7233 ± 0.003
	Management and Development	0.8277 ± 0.021 0.8155 ± 0.013	0.8134 ± 0.015 0.8052 ± 0.013	0.7358 ± 0.015 0.6986 ± 0.002

Table 7: Keywords

Company Abbreviation		
China Unicom	China Telecom	China Mobile
China Broadnet		
Telecom-Related Keywords		
5G	optical fiber	telecommunications
operator	base station	backbone network
gigabit	data center	FTTR
quality of service	roaming	value-added services
service hall	network access	voice services
broadband	cloud computing	local area network
Internet of Things	gateway services	ring back tone
network security	Wi-Fi	core network
network coverage	access network	network optimization
customer relationship management	IPTV	5G plans
data pricing	VoLTE	plan pricing

Do not add, delete, or adjust any tags, attributes, or structure; Only fill in the content of <th> tags – even if a <th> tag has colspan or rowspan attributes, you must still assign it an appropriate column name and must not leave it empty; The return format must be the following JSON structure:

HTML Ranking Prompt

You are an expert in table quality assessment, specializing in analyzing the structure, semantics, and topic relevance of HTML tables. Based on the input below, please evaluate the table's quality across multiple dimensions and provide specific reasons for the scores.

Dimensions (1 to 5, integers only):

Structure Correctness (structure_rank): (The score must be strictly {score})
This score is based only on structural indicators; cell content does not affect this score.
The table's logical column count per row is detailed as follows: {structure_info_prompt}.

Topic Relevance (topic_rank):
It is based only on the relevance between table content and the topic, ignoring expression or semantics.
Assess whether the content matches the topic "{topic}".
Ensure full coverage of the dimensions mentioned in the title (e.g., time, location, metrics, technologies, organizations), no more, no less.
Check whether the table reflects the following entity information: {entities_prompt}. Inclusion of these entities—whether specific or general—is required. Irrelevant or completely unrelated content is prohibited. No fabricated or far-fetched justifications are allowed.
Do not rely solely on the literal presence of keywords; semantic alignment with the topic is sufficient.

Semantic Consistency (semantic_rank):
Are there completely empty cells in the HTML? If so, suggest filling them with content consistent with the table headers.
Are the table headers and body semantically aligned? {complex_string2}
Are the headers clear and detailed?
Are there garbled or corrupted characters?
Deduct points based on the proportion of cells exhibiting the above issues. Do not assign extremely low scores for isolated problems.
Content like "N/A", "-", or "TBD" is not considered empty.

Overall Score (rank):
This should be the lowest score among the three dimensions above.

Output Format should strictly be JSON format, no extra text.
Topic: {topic}
Raw HTML Table Code: {html_code}

Modified CoSyn Prompt

You are a content creation expert with broad domain knowledge. Please generate content material for me based on the following setup:

Topic: {TOPIC}
Figure Type: Table

Please follow the requirements below:

The generated material should be closely related to the topic and tailored according to the persona. The content structure should be suitable for generating the specified figure type (e.g., tables, charts, etc.).

All content must be factual and credible, using real-world entity names. The use of placeholders (such as xxA, xxB, [Name], [Date], etc.) is strictly prohibited.

The material should be diverse and cover the topic from different perspectives to ensure informational richness and breadth.

Control the amount of content and only provide key information so that it fits on a single-page document.

All content must be written in English, even if the persona is non-English-speaking.

Please output in JSON format, do not include any additional explanatory text.

You are an expert in writing HTML documents.
I have a dataset about {TOPIC} that can be used to generate an HTML table.
The data is as follows (in JSON format):
<data>
{DATA}

</data>

Please use HTML and CSS to generate an HTML table. Follow the requirements below:

Styling Requirements:
(1) You may use any CSS frameworks, libraries, or tools to build the page.
(2) Be creative and ensure the webpage is distinctive in terms of typography, colors, borders, layout, etc., while aligning with the topic and target figure type.
(3) Use appropriate design proportions (e.g., margins, page size, content density, etc.) to ensure the information is clearly presented, with no overlapping text or layout issues.
(4) All content must be displayed within a single page—do not make the page too long or too sparse. This is a very important requirement.
(6) All text must be inside the table.
[(7) Generated Table should contain rowspan or colspan attribute.]

Code Requirements:
(1) Please hardcode the provided data directly into the HTML page. Do not use any backend calls. Ensure correct HTML syntax and formatting.
(2) Include both HTML and CSS in a single HTML file. Do not use any external resource files.

Output Requirements:
Please mark the code block.
Do not output any additional explanatory text—the output should be a single HTML file wrapped in <html></html>.

Reproducibility Checklist

Instructions for Authors:
This document outlines key aspects for assessing reproducibility. Please provide your input by editing this `.tex` file directly.
For each question (that applies), replace the "Type your response here" text with your answer.

Example: If a question appears as

```
\question{Proofs of all novel claims are
included} {(yes/partial/no)}
Type your response here
```

you would change it to:

```
\question{Proofs of all novel claims are
included} {(yes/partial/no)}
yes
```

Please make sure to:

- Replace ONLY the "Type your response here" text and nothing else.
- Use one of the options listed for that question (e.g., **yes**, **no**, **partial**, or **NA**).
- **Not** modify any other part of the `\question` command or any other lines in this document.

You can `\input` this .tex file right before `\end{document}` of your main file or compile it as a stand-alone document. Check the instructions on your conference's website to see if you will be asked to provide this checklist with your paper or separately.

---

1. General Paper Structure

1.1. Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/partial/no/NA) yes

1.2. Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes/no) yes

1.3. Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (yes/no) yes

2. Theoretical Contributions

2.1. Does this paper make theoretical contributions? (yes/no) no

If yes, please address the following points:

2.2. All assumptions and restrictions are stated clearly and formally (yes/partial/no) Type your response here

2.3. All novel claims are stated formally (e.g., in theorem statements) (yes/partial/no) Type your response here

2.4. Proofs of all novel claims are included (yes/partial/no) Type your response here

2.5. Proof sketches or intuitions are given for complex and/or novel results (yes/partial/no) Type your response here

2.6. Appropriate citations to theoretical tools used are given (yes/partial/no) Type your response here

2.7. All theoretical claims are demonstrated empirically to hold (yes/partial/no/NA) Type your response here

2.8. All experimental code used to eliminate or disprove claims is included (yes/no/NA) Type your response here

3. Dataset Usage

3.1. Does this paper rely on one or more datasets? (yes/no) yes

If yes, please address the following points:

3.2. A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/NA) yes

3.3. All novel datasets introduced in this paper are included in a data appendix (yes/partial/no/NA) yes

3.4. All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no/NA) yes

3.5. All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations (yes/no/NA) yes

3.6. All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available (yes/partial/no/NA) yes

3.7. All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing (yes/partial/no/NA) NA

4. Computational Experiments

4.1. Does this paper include computational experiments? (yes/no) yes

If yes, please address the following points:

4.2. This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting (yes/partial/no/NA) no

4.3. Any code required for pre-processing data is included in the appendix (yes/partial/no) yes

4.4. All source code required for conducting and analyzing the experiments is included in a code appendix (yes/partial/no) yes

4.5. All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes (yes/partial/no) yes

4.6. All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no) yes

4.7. If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results (yes/partial/no/NA) NA

4.8. This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks (yes/partial/no) partial

4.9. This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics (yes/partial/no) yes

4.10. This paper states the number of algorithm runs used to compute each reported result (yes/no) yes

4.11. Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information (yes/no) yes

4.12. The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank) (yes/partial/no) yes

4.13. This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments (yes/partial/no/NA) yes