TableMix: Enhancing Multimodal Table Reasoning in MLLMs from a Data-Centric Perspective

Supplementary Material

A. Table Perception Data Construction

To enhance the model's ability to perceive and interpret visual table structures, we construct a perception-oriented dataset leveraging both the rendered table images and their corresponding structured tables. This dataset aims to improve the model's low-level perception capabilities, such as recognizing cell boundaries, headers, and alignment, without introducing reasoning or external knowledge.

The primary goal of this sub dataset is to strengthen the model's spatial-structural alignment between the visual layout and the underlying table semantics. Specifically, it teaches the model to (1) accurately localize and read cell content, (2) identify structural components (rows, columns, and merged cells), and (3) recognize formatting cues such as alignment, highlighting. We design seven categories of perception-level questions, automatically derived from structured tables:

- **Cell-level Recognition:** Questions about reading the textual content of specific cells. *Example:* "What is written in the cell located at row 2, column 3?"
- **Header and Structure Understanding:** Questions that test header detection and multi-level structure recognition. *Example:* "What are the column headers in the table?"
- **Counting and Localization:** Tasks involving counting rows, columns, or merged cells. *Example:* "How many rows are there in the table?"
- **Visual-Structural Alignment:** Questions linking visual formatting and structural layout. *Example:* "Are the header cells highlighted or bolded?"
- **Category and Type Recognition:** Tasks identifying data types such as numeric, textual, or percentage. *Example:* "Which column contains percentage values?"
- **Caption and Metadata Recognition:** Questions about auxiliary text elements such as captions or footnotes. *Example:* "What is the title (caption) of the table?"
- **Row/Column Association:** Questions requiring cross-referencing between headers and corresponding cell values. *Example:* "What is the value in the row 'Year 2022' under the column 'Revenue'?"

B. Additional Implementation Details

In this section, we provide additional details of our experimental configuration.

Our training framework is built upon Easy-R1, a lightweight and modular RL fine-tuning system that supports GRPO-based optimization. The temperature for sampling is set to 1.0, and the maximum number of image pixels is capped at 1,881,600 to control GPU memory usage. The maximum feedback length (i.e., model output length during reinforcement learning) is set to 2,000 tokens. The training is running on an 8 H20 GPU node.

For the FeTaQA dataset, we further refine the training data to ensure clean supervision. Specifically, we employ DeepSeek-V3 to re-parse the original question–answer pairs and generate simplified annotations. The following prompt is used:

Prompt for FeTaQA Re-parsing

Process the question and answer appearing in the following JSON, *e.g.*, if there are multiple questions in the question section, break them into separate entries, and in the answer section, keep only the most concise form of the answer. Answer directly, without thinking, in JSON format: `[{"question": "", "answer": ""}]`

This preprocessing step ensures that each instance contains a single, unambiguous question and its corresponding concise answer, which helps reduce noise and improve reward consistency during training.

C. Qualitative Analysis

To provide a more granular understanding of our model's capabilities, we present a qualitative analysis of representative outputs from TableMix, shown in the provided figure. These examples highlight the model's proficiency in handling both multi-step reasoning and complex table structures.

The first case (Figure 8 top) demonstrates a multi-step arithmetic reasoning task. The model is asked to sum the number of calls from two different days. The generated `<think>` process shows that TableMix correctly decomposes the query. It first identifies and extracts the correct call counts for "Tuesday" (4,279) and "Thursday" (1,406) from the simple table. Subsequently, it applies the correct mathematical operation (addition) to arrive at the final, accurate answer. This illustrates the model's ability to construct and execute a logical, sequential plan.

The second case (Figure 8 bottom) presents a more challenging scenario involving a complex, multi-index table with hierarchical rows and columns. The question requires finding the total number of appearances for "Stoke". This task is non-trivial as the table contains multiple "Total" rows

Phone calls received	
Day	Number of calls
Monday	4,560
Tuesday	4,279
Wednesday	2,988
Thursday	1,406
Friday	1,192

Question:
A technical support line tracked how many calls it received each day. How many calls total did the support line receive on Tuesday and Thursday?
TableMix:
<think>\nFirst, identify the number of calls received on Tuesday and Thursday from the table:\n- Tuesday's calls: 4,279\n- Thursday's calls: 1,406\n\nNext, add these two values together to find the total number of calls received on these two days:\n\\[\n4,279 + 1,406 = 5,685\n\\]\n\nThus, the total number of calls received on Tuesday and Thursday is 5,685.\n\n</think>
\\boxed{5685}

Club	Season	League			FA Cup			League Cup			Other			Total		
		Division	Apps	Goals	Apps	Goals		Apps	Goals		Apps	Goals		Apps	Goals	
Stoke City	1968–69	First Division	14	0	0	0		0	0		...			14	0	
	1969–70	First Division	40	4	3	0		1	0		...			44	4	
	1970–71	First Division	36	0	0	2		2	0		0	1		52	3	
	1971–72	First Division	28	5	0	1		0	2		0	2		34	10	
	1972–73	First Division	39	4	1	0		2	0		2	0		44	4	
	1973–74	First Division	41	4	1	0		3	1		3	0		48	5	
	1974–75	First Division	38	2	1	0		5	1		2	1		46	4	
	1975–76	First Division	19	3	1	1		0	0		...			20	4	
	1976–77	First Division	38	2	1	0		0	0		...			39	2	
	1977–78	Second Division	41	1	0	0		1	0		...			42	1	
	1978–79	Second Division	38	2	1	0		3	0		...			44	2	
	1979–80	First Division	34	2	1	0		4	1		...			39	3	
	1980–81	First Division	0	0	0	0		0	0		...			0	0	
	1981–82	First Division	17	0	0	0		2	0		...			19	0	
	Total		407	29	9	4		31	3		23	4		493	52	
York City (loan)	1981–82	Fourth Division	7	1				2	1					9	2	
York City	1982–83	Fourth Division	38	4	4	0		2	0		...			38	4	
	Total		52	5	4	0		5	0		...			55	5	
Career total			644	54	55	4		56	3		25	4		598	67	

Question:
How many appearances did Denis Smith make for Stoke?
TableMix:
<think>\nTo determine the total number of appearances Denis Smith made for Stoke City, we need to look at the \"Total\" row in the \"Stoke City\" section of the table. \n\nThe \"Total\" row for Stoke City shows the following:\n- Apps (appearances): 493\n\nThus, Denis Smith made a total of 493 appearances for Stoke City.\n\n</think>
\\boxed{493}

Figure 8. Qualitative examples from TableMix.

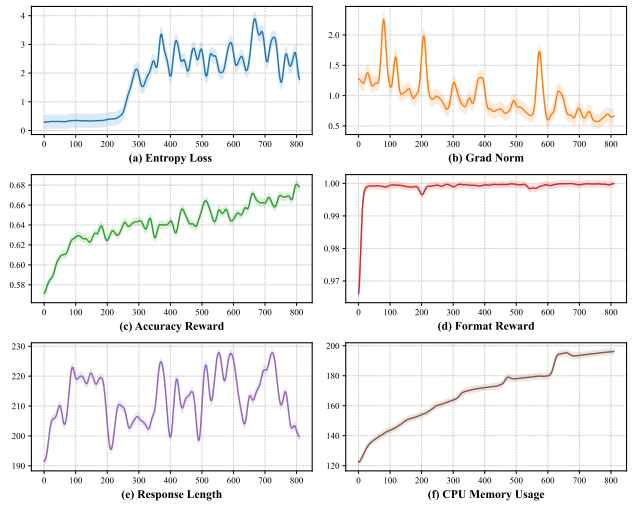

Figure 9. Training dynamics of our proposed model.

(e.g., for "Stoke City," "York City," and "Career total"). The model's reasoning process demonstrates a fine-grained understanding of the table's structure. It correctly identifies that the query refers to the "Total" row specifically within the "Stoke City" club section. It successfully disambiguates this from other summary rows and pinpoints the value in the "Apps" column, yielding the correct 493 appearances.

Collectively, these examples show that TableMix not only performs accurate information retrieval but also constructs logical reasoning plans to navigate complex table structures and execute numerical operations as required by the user's query.

D. Analysis of Training Dynamics

Figure 9 illustrates the training dynamics of our model across six key metrics. The training process reveals distinct learning phases for different objectives.

The Format Reward (d) rapidly saturates at its maximum value early in the training, indicating that the model masters the structural and syntactic requirements of the task almost instantaneously. In contrast, the Accuracy Reward (c), which represents the core semantic task, shows a steady and consistent improvement throughout the entire training process. This disparity highlights that semantic understanding is the primary long-term challenge, while formatting is learned very quickly.

The policy's optimization behavior is captured by the Entropy Loss (a) and Grad Norm (b). The Entropy Loss (a) remains low during the initial training phase before rising sharply to a high, fluctuating plateau. This suggests a clear shift from an initial exploitation phase to a more exploratory policy as training progresses. This increased exploration is corroborated by the high variance exhibited in both the Grad Norm (b) and the Response Length (e). Both metrics show significant oscillations, suggesting the agent is actively navigating a complex optimization landscape rather than settling into a simple, stable policy.

Finally, the CPU Memory Usage (f) shows a steady, almost linear increase throughout the training process, with a slight acceleration in its growth rate during the later stages. This behavior is consistent with an expanding replay buffer or a growing data cache, as expected in our reinforcement learning framework.