A Closer Look into LLMs for Table Understanding

Anonymous ACL submission

Abstract

Despite the success of Large Language Models (LLMs) in table understanding, their internal mechanisms remain unclear. In this paper, we conduct an empirical study on 16 LLMs, covering general LLMs, specialist tabular LLMs, and Mixture-of-Experts (MoE) models, to explore how LLMs understand tabular data and perform downstream tasks. Our analysis focus on 4 dimensions including the attention dynamics, the effective layer depth, the expert activation, and the impacts of input designs. Key findings include: (1) LLMs follow a three-phase attention pattern—early layers scan the table broadly, middle layers localize relevant cells, and late layers amplify their contributions; (2) tabular tasks require deeper layers than math reasoning to reach stable predictions; (3) MoE models activate table-specific experts in middle layers, with early and late layers sharing general-purpose experts; (4) Chain-of-Thought prompting increases table attention, further enhanced by table-tuning. We hope these findings and insights can facilitate interpretability and future research on table-related tasks.

1 Introduction

Tables, as a representative form of structured data, are widely used across various real-world fields to store and present information. In recent years, Large Language Models (LLMs) have shown powerful instruction following and complex reasoning ability, and thus have become the dominant paradigm for table understanding technique, supporting a wide range of downstream application scenarios, such as table question answering (TQA) (Pasupat and Liang, 2015; Zhang et al., 2024; Deng and Mihalcea, 2025), table fact verification (TFV) (Chen et al., 2020; Wang et al., 2024b), advanced data analysis (Li et al., 2025; Weng et al., 2025; OpenAI, 2024) and spreadsheet manipulation (Li et al., 2023; Microsoft, 2025).

Although remarkable success has been achieved with existing LLM-based table understanding approaches, they predominantly focus on how to continually improve performance metrics across downstream tasks, e.g., boosting LLMs' performance with the most suitable prompt designs (Sui et al., 2024; Ye et al., 2023) and developing specialist tabular LLMs with supervised fine-tuning (SFT) (Deng and Mihalcea, 2025; Zhang et al., 2024, 2025) or reinforcement learning (RL) (Yang et al., 2025b; Wu et al., 2025; Kang et al., 2025b). Unlike previous performance-oriented work, in this paper, we conduct an in-depth empirical study to achieve a deeper understanding of the underlying mechanisms of LLMs' table understanding. Our experiments cover a wide spectrum of LLMs of different sizes and types, and focuses on four perspectives including the attention dynamics, the effective layer depth, the impact of MoE architectures and the influence of input designs. The explored research questions are shown in below.

Q1: How do intra-table and extra-table attention patterns of LLMs evolve when performing table understanding tasks?

Q2: How many effective layers are utilized by LLMs to achieve stable predictions and are there differences between tabular and general tasks?

Q3: Are there specialized table experts for MoE models and where are they located?

Q4: The influence of different table formats and reasoning strategies on internal representations.

We conclude our key findings as follows:

(1) Different LLM layers exhibit distinct attention patterns during table understanding. The early layers broadly scan the overall table content, the middle layers then concentrate attention on query-related cells and the top layers further amplify the focused content representations to produce the final answer.

(2) Tabular tasks require more layers for output distribution refinement than math reasoning.

While the depth at which final answer content crystallizes is broadly comparable across tasks, LLMs refine output distributions across more layers when processing tabular data—a pattern consistent across model scales and training strategies.

(3) MoE models develop specialized experts for tabular tasks in the middle layers, which show minimal overlap with math-oriented experts, but early and top layers share general-purpose experts across different tasks.

(4) Input formats and reasoning strategies indeed influence attention patterns. HTML tables result in more dispersed attention distributions in early layers than Markdown tables. Chain-of-Thought (CoT) prompting can allocate more attention to the table content, which can be further amplified by table-specific fine-tuning and thus leads to better performance.

We believe that our findings not only draw new insights into LLMs' interpretability in tabular domain, but also serve as a valuable guideline for future research—ranging from optimal input configurations that maximize attention to relevant table content, to inference-time interventions targeting later layers, to MoE-specific optimizations on table-specialized experts.

2 Related Work

Our work is situated at the intersection of LLM-based table understanding (TU) and LLM interpretability. Prior TU research has sought to improve LLM performance on tabular tasks through prompt engineering (Chen, 2023; Ye et al., 2023; Wang et al., 2024b; Ji et al., 2024), instruction tuning (Zhang et al., 2024, 2025), reinforcement learning (Yang et al., 2025b; Wu et al., 2025; Zhou et al., 2025a), and agent-based frameworks (Li et al., 2023; Weng et al., 2025; Wang et al., 2025), while robustness studies have examined sensitivity to table formats (Sui et al., 2024), noisy operators (Singha et al., 2023), and cell perturbations (Bhandari et al., 2025). On the interpretability side, existing work has investigated layer-wise information processing (Skean et al., 2025; Csordás et al., 2025; Hu et al., 2025), the roles of attention heads and FFN modules in knowledge representation and retrieval (Geva et al., 2021; Meng et al., 2022; Wu et al., 2024), phenomena such as attention sinks (Xiao et al., 2023; Gu et al., 2024; Qiu et al., 2025), and expert specialization in MoE architectures (Wang et al., 2024a; Bai et al., 2025). However, these interpretability efforts predominantly target unstructured text tasks like math reasoning and factual recall. In contrast, our work extends interpretability analysis to structured tabular data, aiming to uncover the internal mechanisms by which LLMs process tables and to provide actionable insights for future research. A more comprehensive discussion of related work is provided in Appendix F.

3 Experimental Setups

Evaluation Data We randomly select 500 samples from 3 TQA benchmarks and 1 TFV benchmark to perform empirical study, which include WTQ (Pasupat and Liang, 2015), HiTab (Cheng et al., 2021), AIT-QA (Katsis et al., 2021) and TabFact (Chen et al., 2020). The resulting 2,000 test samples cover common table structures with flat headers (WTQ, TabFact) and complex hierarchical headers (HiTab, AIT-QA). The input tables are serialized into Markdown format by default, and we also analyze the impact of alternative formats (e.g., HTML) in Section 7.1. The complete information, such as input prompt templates and the overall statistics, is shown in Appendix C.1 and C.3.

Analyzed LLMs We analyze 16 LLMs of three categories. (1) general-purpose instruct LLMs such as Llama-3.1-8B-Instruct and Qwen-2.5-7B-Instruct. (2) fine-tuned tabular LLMs via SFT or RL such as TAMA (Deng and Mihalcea, 2025) and Table-R1-Zero (Yang et al., 2025b). (3) Mixture-of-Experts (MoE) models like DeepSeek-V2-Lite (DeepSeek-AI et al., 2024) and Qwen3-30B-A3B (Yang et al., 2025a). The complete model list is provided in Appendix C.4.

4 Analysis of Attention Dynamics

To understand how LLMs allocate attention during table question answering, we design a controlled experimental setup where each input instance comprises three segments: (1) a **system prompt** providing general instructions, (2) a **table content** containing the serialized table, and (3) a **user question** specifying the query. This segmentation allows us to trace attention flow across input components. We investigate three complementary questions: **§4.1**: How does the model attend to three input segments, especially different table cells? **§4.2**: How much does table content actually contribute to the model's final output? **§4.3**: Does this attention pattern causally affect model predictions?

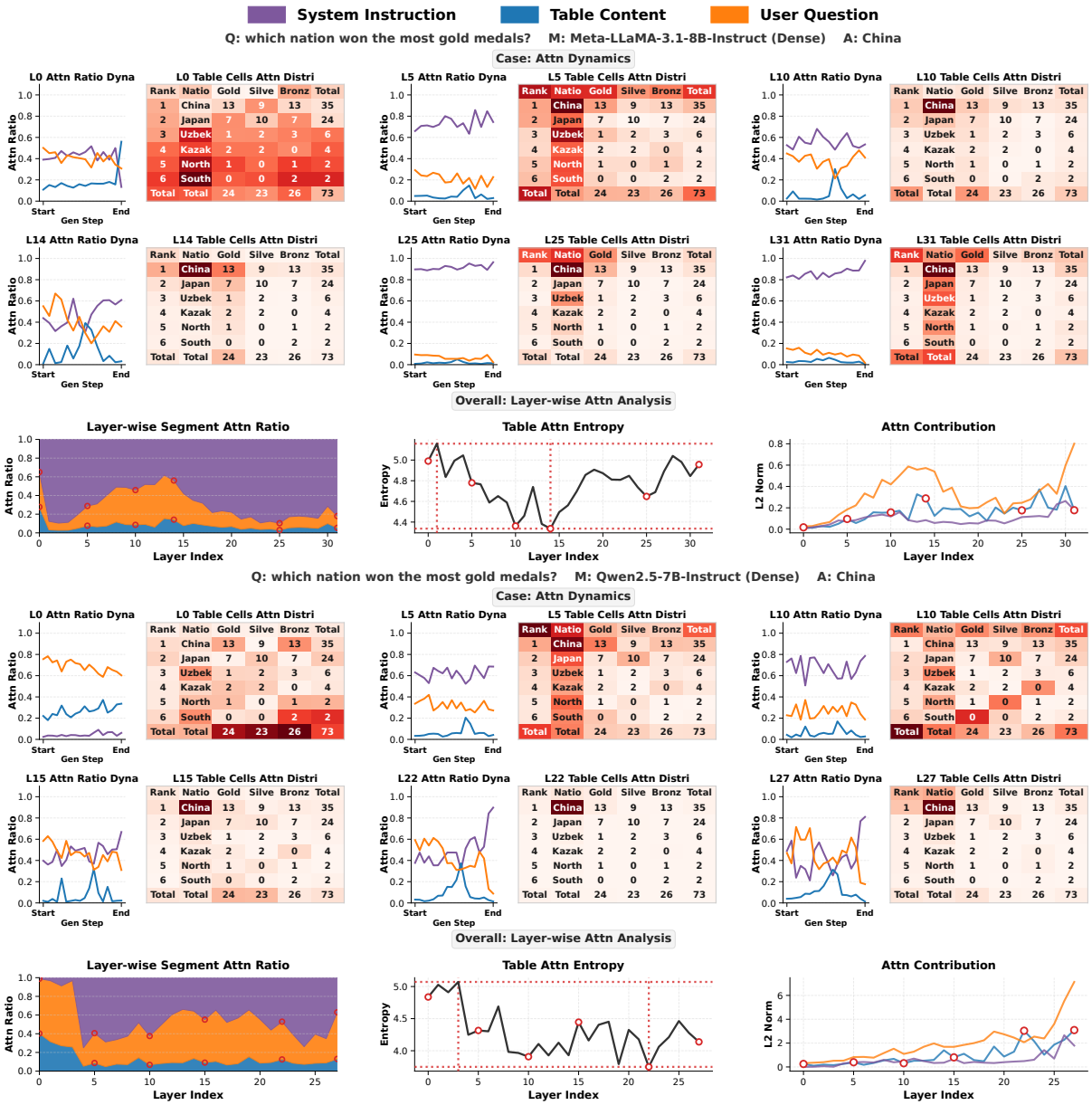

Figure 1: The attention dynamics of Llama3.1-8B-Instruct and Qwen2.5-7B-Instruct on tabular tasks. Each input consists of three segments: **system prompt**, **table content**, and **user question**. **Upper**: A case study at selected layers—left sub-panels show step-wise attention ratio trend throughout generation steps; right sub-panels show cell-level attention heatmaps. **Lower**: Aggregated results—*Layer-wise Segment Attn Ratio* shows the proportion of attention allocated to each segment per layer; *Table Attn Entropy* measures the degree of focus towards table cells, with lower entropy indicating a more concentrated attention distribution on specific cells); *Attn Contribution* measures the influence of each segment on the final output.

The following metrics are employed in our analysis. (1) *segment attention ratio* measures the proportion of attention allocated to each segment per layer; (2) *table attention entropy* quantifies how concentrated the attention is within the table (lower entropy indicates a more focused attention distribution on specific cells); and (3) *attention contribution* captures the actual influence of each segment on the final output via L2 norm of value-weighted representations. Formal definitions and formulas of these metrics are provided in Appendix B. For all metrics, we first average across all generated tokens of each sample, and then average across all 2,000 test samples. The attention dynamics of Llama3.1-8B-Instruct and Qwen2.5-7B-Instruct is shown in Figure 1. Results of more models are shown in Appendix G.

4.1 How Does the Model Attend to the Table?

Takeaway 1

Although the overall attention ratio to table content is modest compared to other segments, LLMs can precisely focus on query-relevant cells in the middle layers, suggesting that accurate localization matters more than total attention volume.

Allocated attention ratio across input segments. As shown in Figure 1 (lower left), the attention allocated to **<table content>** is highest in the earliest layers, then it drops before rising again to a secondary peak in the lower-middle layers (layer 10-15) and finally decreases to a stable level in the top layers (layer 20-30). Notably, the overall changes of table-oriented attention across different layers are modest. We observe a similar trend of attention allocated to **<user question>**, with both reaching their peak in the middle layers, which may suggest that the model is integrating question information to interpret table structures and identify critical table content. Except for middle layers, the **<system prompt>** consistently attracts relatively higher attention scores across all layers than user question and table content, which could be attributed to the attention sink phenomenon observed in prior work (Xiao et al., 2023; Gu et al., 2024; Kang et al., 2025a). The step-wise segment attention ratio in specific layers throughout the token-by-token generation process of one sample is also visualized in the upper left sub-panels in Figure 1, which provides evidence of observed patterns.

The influence of model scales. Comparing models of different scales within the same family (detailed results in Appendix G), we find that overall attention distribution trends remain consistent. However, two notable differences emerge: (1) the secondary peak of table attention shifts toward deeper layers as model size increases, and (2) the layer range with low table attention entropy becomes broader in larger models, suggesting they maintain focused attention across more layers.

The influence of model architectures. Comparing Llama3.1-8B and Qwen2.5-7B (Figure 1), we observe that Qwen allocates higher attention to table content in early layers. MoE models exhibit a more gradual decline in table attention entropy, with a distinctive three-phase pattern: attention first covers partial cells, then spreads across the entire table, and finally converges to answer-relevant cells (detailed in §6). For tabular LLMs like TableGPT2 (Su et al., 2024), table-specific fine-tuning leads to higher table attention and weaker attention sink effects, though overall patterns remain similar to their base models.

Attention distribution within the table. While the overall attention allocated to table is limited, does the model at least focus on the query-related table cells? To answer this, we measure the *table attention entropy* over table tokens at each layer.

As shown in Figure 1 (lower middle), the table attention entropy follows a U-shaped pattern: it decreases from early layers, and reaches a minimum in the middle layers, and finally increases again. The attention heatmaps in Figure 1 (upper right sub-panels) also illustrate this pattern. In early layers (e.g., Layer 0), attention is spread broadly across the entire table, suggesting that the model scans all content. In middle layers (e.g., Layer 14 for LLaMA-3.1-8B-Instruct), attention becomes concentrated on specific cells—for the question "which nation won the most gold medals?", the model focuses on the "China" cell and the "Gold" column header, which directly correspond to the answer. In later layers, attention spreads slightly again as the model prepares to generate the answer. This pattern is consistent across different models and benchmarks. We provide more case studies and model comparisons in the Appendix G.

These findings suggest that LLMs selectively focus on query-relevant cells rather than attending to all table content uniformly, implying that low-attention table regions could be pruned to develop more efficient tabular LLMs with minimal performance loss.

4.2 How Much Does the Table Content Contribute to the Final Output?

Takeaway 2

Despite receiving modest attention, table content makes substantial contributions to the model's output, with its influence increasing in later layers and ultimately driving answer generation.

Attention weights show where the model "looks", but not how much the attended content actually influences the output. As noted by prior work (Kobayashi et al., 2020; Gu et al., 2024), the

true impact depends on both the attention weight and the transformed value vector. Therefore, we measure the *value-weighted contribution*—the L2 norm of each segment's representation injected into the transformer residual stream (see Appendix B for formal definition).

As shown in Figure 1 (lower right), despite the modest attention ratio observed in §4.1, the table segment makes substantial contributions that increase in later layers, peaking near the final layers. In contrast, the system prompt's contribution remains low across all layers despite its high attention ratio, confirming its role as an attention sink rather than an information source.

Together, these findings reveal a coherent three-phase workflow: early layers broadly encode the table, middle layers focus on query-relevant cells and amplify their residual-stream impact, and late layers use the focused table content with the question to drive answer generation.

4.3 Does the Attention Pattern Causally Affect Output?

Takeaway 3

The three-phase attention pattern is functionally essential rather than a mere processing correlate: masking table attention in early-to-middle layers causes severe performance degradation, and middle-layer attention focus on answer cells is predictive of output correctness.

Causal validation via attention masking. We zero out attention weights to all table tokens within consecutive quarter-layer blocks—Shallow (1–L/4), S-M (L/4–L/2), M-D (L/2–3L/4), and Deep (3L/4–L)—across 5 models, 3 benchmarks, and 2 formats. As shown in Figure 2, masking Shallow and S-M layers causes the largest performance drops (44–65 points), confirming that early broad scanning is functionally essential; Llama-family models are most sensitive to early masking and least sensitive to Deep masking, while Qwen-family models show more uniform degradation across all ranges, reflecting their more distributed attention profiles (§4.1).

We further perform manual verification at scale and validate the correlation between middle-layer attention focus and output quality, with detailed statistics provided in Appendix E.2.

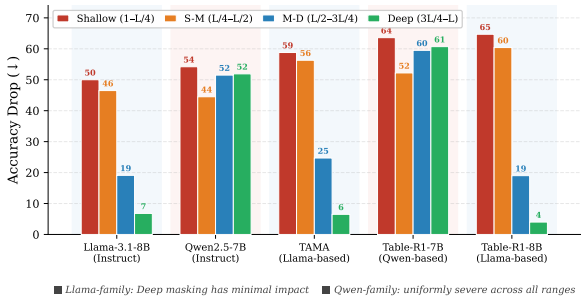

Figure 2: Performance drop when masking table attention in different layer ranges across 5 models (averaged over 3 benchmarks × 2 formats).

5 Effective Depth for Tabular Tasks

Takeaway 4

Table tasks engage more layers for output distribution refinement than math reasoning, though the depth at which answer content crystallizes is comparable across tasks. This pattern is consistent across model scales and fine-tuning strategies, suggesting that layer-wise functionality is largely determined during pre-training.

The previous analysis revealed that middle layers concentrate attention on query-relevant cells while later layers amplify their contributions. A natural follow-up question is: how many layers do LLMs actually need to complete table understanding tasks? To investigate this, we apply Logit Lens (Nostalgebraist., 2020), which decodes hidden representations at each layer into vocabulary distributions. We measure the KL divergence between each layer's prediction and the final output, along with the top-5 token overlap.

As shown in Figure 3, KL divergence remains relatively high throughout the middle layers and only drops sharply in final layers. To directly validate whether this pattern distinguishes table tasks from other reasoning tasks, we run the same Logit Lens pipeline on GSM8K across 7 representative models spanning general instruction-tuned, table-specific fine-tuned, and MoE categories. We decompose stabilization depth into two complementary metrics: *KL-based depth* (where output distribution converges) and *Top-k overlap depth* (where the final answer content crystallizes).

As shown in Table 1, KL-based stabilization depth is consistently higher for table tasks than math in most models (5 of 7, marked with †), indicating that LLMs refine their output distributions

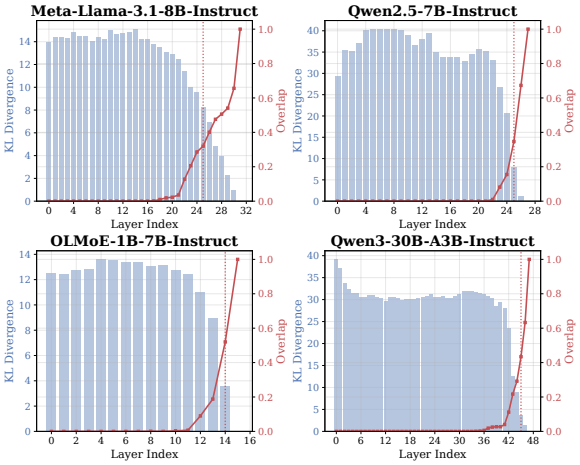

Figure 3: Prediction stability analysis using LogitLens. Bars show KL divergence between each layer's decoded distribution and the final output (lower = closer to final prediction). Lines show top-5 token overlap with the final output. Vertical dashed lines mark where predictions stabilize.

across more layers for tabular data. By contrast, Top-k overlap depth is broadly similar across task types, suggesting that final answer content crystallizes at a comparable stage. Together, these results suggest that table understanding does not simply take longer overall, but specifically requires more layers for fine-grained distribution refinement after answer identification.

Notably, this pattern holds across different model scales (7B to 32B) and training strategies (SFT and RL). This finding has practical implications: unlike math reasoning, where early-exit strategies can reduce computation (Csordás et al., 2025; Hu et al., 2025), table tasks may benefit less from such optimizations. Conversely, targeted interventions in later layers (e.g., inference-time steering) could be especially effective for improving table understanding (detailed results in Appendix D.2).

6 Seeking Table Experts of MoE Models

Takeaway 5

MoE models activate a distinct set of table-specific experts concentrated in the middle layers, mirroring the attention patterns observed in RQ1. These experts show minimal overlap with math-related experts.

Mixture-of-Experts (MoE) models route each token to a subset of experts based on learned gating functions. A natural question arises: do MoE models develop specialized experts for tabular tasks?

Table 1: Stabilization depth comparison between table and math (GSM8K) tasks. Values are reported as effective layer / total layers (ratio). KL-based depth is higher for table tasks in 5 of 7 models (†), while Top-k overlap depth is broadly comparable.

Model	Layers	KL Depth		Overlap Depth	
		Table	Math	Table	Math
Instruct LLMs					
Qwen2.5-3B	36	32 (0.89)†	27 (0.75)	33 (0.92)	31 (0.86)
Qwen2.5-7B	28	25 (0.89)†	23 (0.82)	25 (0.89)	25 (0.89)
Qwen2.5-14B	48	44 (0.92)†	43 (0.90)	44 (0.92)	43 (0.90)
Llama-3.1-8B	32	26 (0.81)	29 (0.91)	25 (0.78)	29 (0.91)
Table-Specific Fine-tuned LLMs					
TAMA	32	26 (0.81)†	23 (0.72)	25 (0.78)	26 (0.81)
Table-R1-7B	28	25 (0.89)†	24 (0.86)	25 (0.89)	25 (0.89)
MoE LLMs					
Qwen3-30B-A3B	48	43 (0.90)	43 (0.90)	45 (0.94)	43 (0.90)

To identify table-specific experts, we record, on the same set described in Section 3, the activation frequency of each expert across all table tokens and layers, following prior work (Wang et al., 2024a; Bai et al., 2025). As shown in Figure 4 (left), table-relevant experts are distributed across all layers, but their activation patterns vary by depth. The middle column of Figure 4 reveals a striking pattern: the entropy of expert activation distribution reaches its minimum in the middle layers across all three MoE architectures. Lower entropy indicates that the model concentrates its routing on fewer, more specialized experts. This pattern mirrors the attention entropy findings in RQ1, where attention within table also becomes most concentrated in the middle layers. Furthermore, these table-specific experts in middle layers show minimal overlap with experts activated for math reasoning tasks (Figure 4, right), confirming their domain-specific nature.

Together, these findings suggest that MoE models achieve efficient multi-task processing by sharing general-purpose experts in early and late layers while routing to specialized experts in the middle layers for domain-specific reasoning.

7 Impact of Input Formats and Reasoning Strategies

The previous sections analyzed LLMs' internal mechanisms under a fixed setting (Markdown format, direct answering). In practice, however, tables can be represented in various formats (e.g., Markdown, HTML), and models can be prompted with different reasoning strategies (e.g., direct answering, Chain-of-Thought). Here we examine how these choices influence internal processing and model behavior.

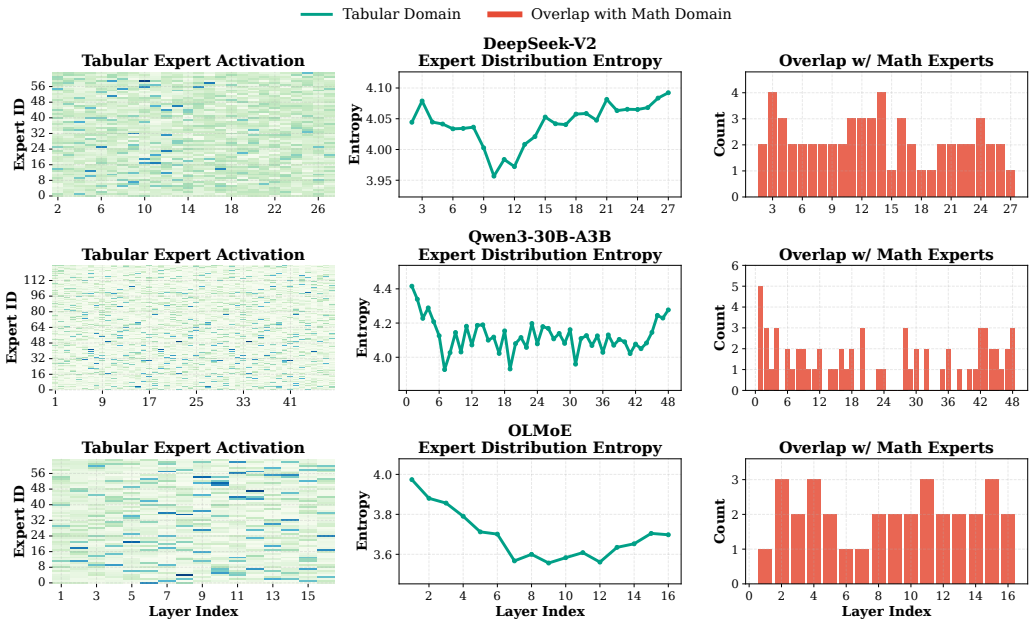

Figure 4: Expert activation analysis in MoE models across three architectures (DeepSeek-V2, Qwen3-30B-A3B, OLMoE). **Left**: Activation heatmaps showing table-specific experts per layer (darker = higher activation probability). **Middle**: Entropy of expert activation distribution for table tasks (lower entropy indicates concentrated activation on fewer experts). **Right**: Number of overlapping experts between table and math (GSM8K) tasks per layer.

7.1 How Does Table Format Affect Attention?

Takeaway 6

HTML tables lead to higher attention entropy than Markdown tables. However, internal representations of different formats converge in deeper layers.

To investigate how table format affects internal processing, we convert our analysis data from Markdown to HTML format and compare the attention patterns. As shown in Figure 5, LLMs assign a slightly higher attention ratio to HTML tables, but the attention entropy is significantly higher. This may be because the verbose structural syntax in HTML (e.g., `<tr>`, `<td>` tags) disperses the model's attention across more tokens, making it harder to concentrate on specific cells.

Despite these differences in early layers, t-SNE visualizations reveal that the internal representations of different formats gradually converge as layer depth increases. This suggests that while early layers process format-specific features, deeper layers extract format-agnostic semantic information. This convergence may explain the minimal performance gap between formats observed in Table 2, despite their different syntactic structures.

Table 2: Average performance over 2,000 test samples under different input formats and inference strategies.

	Markdown		HTML	
Model	Vanilla	CoT	Vanilla	CoT
Instruct LLMs				
Qwen2.5-7B	62.15	67.20	61.35	67.80
Llama3.1-8B	56.95	62.35	57.95	58.35
Table-Specific Fine-tuned LLMs				
Table-R1-7B	69.75	**82.30**	70.30	**81.60**
Table-R1-8B	70.85	82.25	69.90	80.60
MoE LLMs				
Qwen3-30B-A3B	66.30	70.15	68.35	72.10

7.2 How Does Reasoning Strategy Affect Attention?

Takeaway 7

Chain-of-Thought (CoT) reasoning leads to higher attention to table content in middle and late layers compared to direct answering, an effect that persists even when analyzing only final answer tokens. Table-specific fine-tuning further amplifies this effect.

We compare the attention patterns under direct answering and Chain-of-Thought (CoT) prompting for both vanilla LLMs and their table-tuned

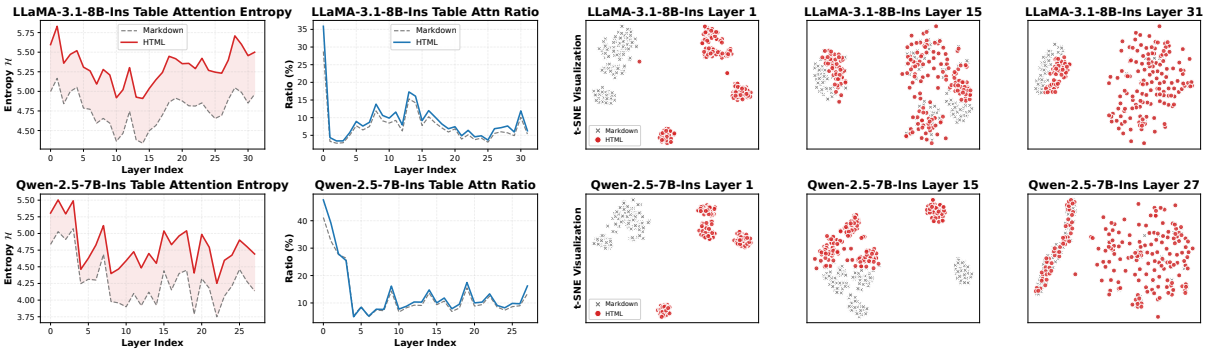

Figure 5: Comparison between Markdown and HTML table formats.

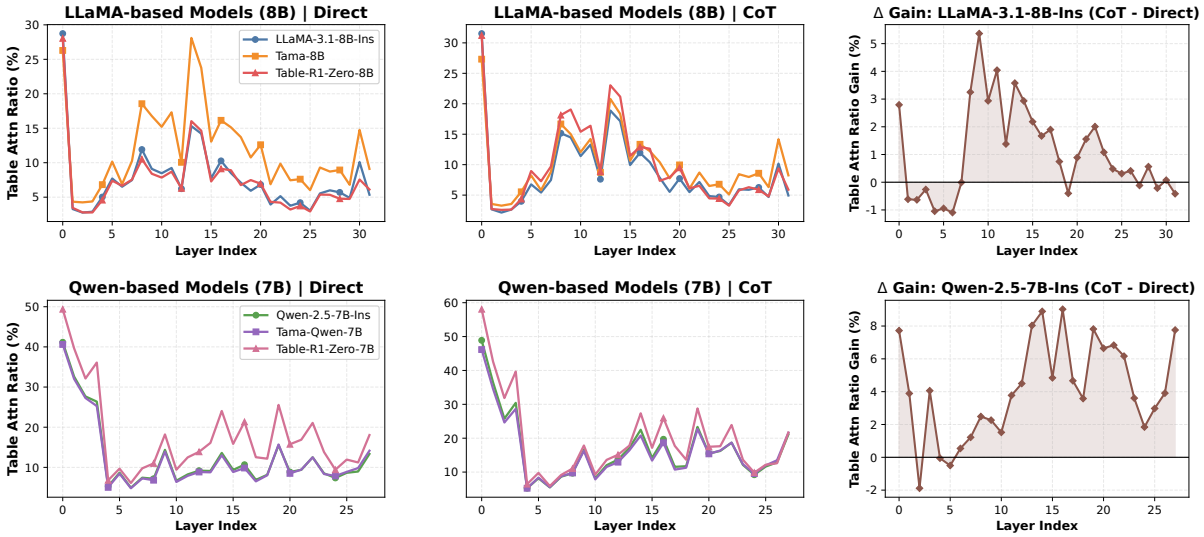

Figure 6: Comparison between direct answering and Chain-of-Thought (CoT) reasoning.

variants. As shown in Figure 6 (bottom), CoT reasoning leads to a higher proportion of attention allocated to table content, especially in the middle and late layers. Comparing vanilla and table-tuned models (Figure 6, top), table-specific fine-tuning generally increases attention to tables across all layers, with RL-tuned models showing the largest increase when combined with CoT. The performance results in Table 2 corroborate this: CoT consistently improves accuracy across all models, with particularly striking gains for table-tuned models (e.g., Table-R1-7B improves from 69.75% to 82.30%).[1]

To further isolate CoT's effect from within-chain table references, we recompute the Table Attn Ratio using *attention only at final answer tokens*. CoT still yields higher early-layer table attention under this restricted setting, confirming that it strengthens table grounding during answer generation. In middle-to-late layers, however, the ratio under CoT falls slightly below direct answering: the reasoning chain has already distilled relevant table information into the residual stream, allowing later layers to attend to the nearby reasoning trajectory rather than re-attending to distant table tokens.

[1]All CoT vs. Direct differences are statistically significant; complete results with 95% bootstrap confidence intervals are in Appendix D.1.

8 Conclusion

We present the first systematic study of the internal mechanisms underlying LLM-based table understanding across 16 LLMs. Our analysis reveals a three-phase workflow: early layers broadly encode the table, middle layers localize query-relevant cells, and late layers amplify focused content for answer generation. This pattern holds across architectures—MoE models mirror it through table-specific expert activation in middle layers, while CoT reasoning and table-specific fine-tuning further enhance table engagement. Together, these findings provide practical guidance for input design, later-layer interventions, and targeted optimization of table-specialized experts.

9 Limitations

Though this paper presents a systematic empirical study of the internal mechanisms of LLM-based table understanding, there are certain limitations and promising directions that deserve future research. (1) Analyzing LLMs of larger scales and different series. Due to resource limitations, we mainly analyze advanced open-source models with parameters below 32B. Models of large scales such as DeepSeek-V3-671B and Qwen3-235B-A22B also deserve further investigation to analyze the scaling effect on table understanding. (2) Extending to MLLMs. Previous work and recent proprietary models have demonstrated that multimodal LLMs (MLLMs) also possess strong table understanding ability based on table images (Zheng et al., 2024; Deng et al., 2024; Zhou et al., 2025b). As a result, it is worthwhile to extend our analysis to MLLM scope to discover unique characteristics of multimodal table understanding. (3) Analyzing more tabular tasks. This study primarily focuses on table question answering and table fact verification, two most typical tasks of existing studies. However, it is pertinent to investigate whether our findings can generalize to more complex tabular tasks, such as data analysis and table summarization. (4) Enriching table formats and reasoning strategies. We do not intend to exhaust every possible table formats such as CSV and JSON formats and every reasoning strategies such as program-of-thoughts with in-context examples, which could be further explored by future follow-ups.

10 Ethical Considerations

Our empirical study and all experiments are conducted based 4 public academic benchmarks, which are free and open-source data for research use. The analyzed LLMs are also open-source models that we downloaded from their official websites. As a result, the authors foresee no ethical concerns.

References

Jun Bai, Minghao Tong, Yang Liu, Zixia Jia, and Zilong Zheng. 2025. Understanding and leveraging the expert specialization of context faithfulness in mixture-of-experts LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21938–21953, Suzhou, China. Association for Computational Linguistics.

Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun Peng. 2025. Multilingual routing in mixture-of-experts. *arXiv preprint arXiv:2510.04694*.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255, Vienna, Austria. Association for Computational Linguistics.

Kushal Raj Bhandari, Sixue Xing, Soham Dan, and Jianxi Gao. 2025. Exploring the robustness of language models for tabular question answering via attention analysis. *Preprint*, arXiv:2406.12719.

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, Dubrovnik, Croatia. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. *Preprint*, arXiv:1909.02164.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2021. Hitab: A hierarchical table dataset for question answering and natural language generation. *arXiv preprint arXiv:2108.06712*.

Róbert Csordás, Christopher D Manning, and Christopher Potts. 2025. Do language models use their depth efficiently? *arXiv preprint arXiv:2505.13898*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, and 81 others. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *CoRR*, abs/2405.04434.

Naihao Deng and Rada Mihalcea. 2025. Rethinking table instruction tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21757–21780, Vienna, Austria. Association for Computational Linguistics.

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. 2024. Tables as texts or images: Evaluating the table reasoning ability of LLMs and MLLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 407–426, Bangkok, Thailand. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2024. When attention sink emerges in language models: An empirical view. *arXiv preprint arXiv:2410.10781*.

Yi Hu, Cai Zhou, and Muhan Zhang. 2025. What affects the effective depth of large language models? *arXiv preprint arXiv:2512.14064*.

Deyi Ji, Lanyun Zhu, Siqi Gao, Peng Xu, Hongtao Lu, Jieping Ye, and Feng Zhao. 2024. Tree-of-table: Unleashing the power of llms for enhanced large-scale table understanding. *Preprint*, arXiv:2411.08516.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023. StructGPT: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251, Singapore. Association for Computational Linguistics.

Seil Kang, Jinyeong Kim, Junhyeok Kim, and Seong Jae Hwang. 2025a. See what you are told: Visual attention sink in large multimodal models. *arXiv preprint arXiv:2503.03321*.

Xiaoqiang Kang, Shengen Wu, Zimu Wang, Yilin Liu, Xiaobo Jin, Kaizhu Huang, Wei Wang, Yutao Yue, Xiaowei Huang, and Qiufeng Wang. 2025b. Can GRPO boost complex multimodal table understanding? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12631–12644, Suzhou, China. Association for Computational Linguistics.

Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan, Jaydeep Sen, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2021. Ait-qa: Question answering dataset over complex tables in the airline industry. *Preprint*, arXiv:2106.12944.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2023. Analyzing feed-forward blocks in transformers through the lens of attention maps. *arXiv preprint arXiv:2302.00456*.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and Zhaoxiang Zhang. 2023. Sheetcopilot: Bringing software productivity to the next level through large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jinyang Li, Nan Huo, Yan Gao, Jiayi Shi, Yingxiu Zhao, Ge Qu, Bowen Qin, Yurong Wu, Xiaodong Li, Chenhao Ma, Jian-Guang Lou, and Reynold Cheng. 2025. Are large language models ready for multi-turn tabular data analysis? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 34795–34835. PMLR.

Ka Man Lo, Zeyu Huang, Zihan Qiu, Zili Wang, and Jie Fu. 2025. A closer look into mixture-of-experts in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4427–4447.

Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. 2025. Large language model for table processing: a survey. *Frontiers of Computer Science*, 19(2).

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Microsoft. 2025. Copilot in excel. Accessed: 2025-12-29.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, and 5 others. 2024. Olmoe: Open mixture-of-experts language models. *CoRR*, abs/2409.02060.

Nostalgebraist. 2020. Interpreting gpt: the logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens.

OpenAI. 2024. Improvements to data analysis in chatgpt. Accessed: 2025-12-29.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Zihan Qiu, Zekun Wang, Bo Zheng, Zeyu Huang, Kaiyue Wen, Songlin Yang, Rui Men, Le Yu, Fei Huang, Suozhi Huang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. Gated attention for large language models: Non-linearity, sparsity, and attention-sink-free. *Preprint*, arXiv:2505.06708.

Enrique Queipo-de Llano, Álvaro Arroyo, Federico Barbero, Xiaowen Dong, Michael Bronstein, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Attention sinks and compression valleys in llms are two sides of the same coin. *arXiv preprint arXiv:2510.06477*.

Alexey Shigarov. 2023. Table understanding: Problem overview. *WIREs Data Mining and Knowledge Discovery*, 13(1):e1482.

Qingyi Si, Tong Wang, Zheng Lin, Xu Zhang, Yanan Cao, and Weiping Wang. 2023. An empirical study of instruction-tuning large language models in Chinese. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4086–4107, Singapore. Association for Computational Linguistics.

Ananya Singha, José Cambronero, Sumit Gulwani, Vu Le, and Chris Parnin. 2023. Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms. *Preprint*, arXiv:2310.10358.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*.

Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Gang Chen, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, Haoze Li, Haoxuan Lan, Jiaming Tian, Jing Yuan, Junbo Zhao, Junlin Zhou, Kaizhe Shou, Liangyu Zha, Lin Long, and 14 others. 2024. Tablegpt2: A large multimodal model with tabular data integration. *Preprint*, arXiv:2411.02059.

Zunhai Su, Qingyuan Li, Hao Zhang, Weihao Ye, Qibo Xue, YuLei Qian, Yuchen Xie, Ngai Wong, and Kehong Yuan. 2025. Unveiling super experts in mixture-of-experts large language models. *arXiv preprint arXiv:2507.23279*.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. *Preprint*, arXiv:2305.13062.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Zihan Wang, Deli Chen, Damai Dai, Runxin Xu, Zhuoshu Li, and Yu Wu. 2024a. Let the expert stick to his last: Expert-specialized fine-tuning for sparse architectural large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 784–801.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. 2024b. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *ICLR*.

Ziwei Wang, Jiayuan Su, Mengyu Zhou, Huaxing Zeng, Mengni Jia, Xiao Lv, Haoyu Dong, Xiaojun Ma, Shi Han, and Dongmei Zhang. 2025. Sheetbrain: A neuro-symbolic agent for accurate reasoning over complex and large spreadsheets. *Preprint*, arXiv:2510.19247.

Han Weng, Zhou Liu, Yuanfeng Song, Xiaoming Yin, Xing Chen, and Wentao Zhang. 2025. Unidatabench: Evaluating data analytics agents across structured and unstructured data. *Preprint*, arXiv:2511.01625.

Wenhao Wu, Yizhong Wang, Guangxuan Xiao, Hao Peng, and Yao Fu. 2024. Retrieval head mechanistically explains long-context factuality. *arXiv preprint arXiv:2404.15574*.

Zhenhe Wu, Jian Yang, Jiaheng Liu, Xianjie Wu, Changzai Pan, Jie Zhang, Yu Zhao, Shuangyong Song, Yongxiang Li, and Zhoujun Li. 2025. Table-r1: Region-based reinforcement learning for table understanding. *Preprint*, arXiv:2505.12415.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Shang Yang, Haotian Tang, Yao Fu, Song Han, and 1 others. Duoattention: Efficient long-context llm inference with retrieval and streaming heads. In *The Thirteenth International Conference on Learning Representations*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Zheyuan Yang, Lyuhao Chen, Arman Cohan, and Yilun Zhao. 2025b. Table-r1: Inference-time scaling for table reasoning tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20616–20635, Suzhou, China. Association for Computational Linguistics.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 174–184, New York, NY, USA. Association for Computing Machinery.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024. TableLlama: Towards open large generalist models for tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6024–6044, Mexico City, Mexico. Association for Computational Linguistics.

Xiaokang Zhang, Sijia Luo, Bohan Zhang, Zeyao Ma, Jing Zhang, Yang Li, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, Jifan Yu, Shu Zhao, Juanzi Li, and Jie Tang. 2025. TableLLM: Enabling tabular data manipulation by LLMs in real office usage scenarios. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10315–10344, Vienna, Austria. Association for Computational Linguistics.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. 2024. Multimodal table understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9102–9124, Bangkok, Thailand. Association for Computational Linguistics.

Wei Zhou, Mohsen Mesgar, Heike Adel, and Annemarie Friedrich. 2025a. p2-tqa: A process-based preference learning framework for self-improving table question answering models. *Preprint*, arXiv:2505.17565.

Wei Zhou, Mohsen Mesgar, Heike Adel, and Annemarie Friedrich. 2025b. Texts or images? a fine-grained analysis on the effectiveness of input representations and models for table question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2307–2318, Vienna, Austria. Association for Computational Linguistics.

A Background

A.1 Task Formalization

Given a table $\mathcal{T}$ and a natural language question $\mathcal{Q}$, Table Question Answering (TableQA) requires the model $G(\cdot)$ to generate a final answer $\mathcal{A}$ based on the above information. Within the context of Large Language Models (LLMs), the input sequence x is composed of three segments serving distinct semantic roles:

- **System Prompt Segment** ($\mathcal{S}_{sys}$): Comprises the system instructions and model-specific conversation templates (e.g., `<|begin_of_text|>` for Llama-3.1-8B-Instruct (Dubey et al., 2024)).
- **Table Content Segment** ($\mathcal{S}_{tab}$): Contains the table data serialized in a specific format (e.g., Markdown or HTML).
- **Question Description Segment** ($\mathcal{S}_{qst}$): Contains the specific user query $\mathcal{Q}$.

Let $\mathcal{I}_{sys}$, $\mathcal{I}_{tab}$, and $\mathcal{I}_q$ denote the sets of indices corresponding to these segments in the input sequence, the total input index set is $\mathcal{I}_{in} = \mathcal{I}_{sys} \cup \mathcal{I}_{tab} \cup \mathcal{I}_q$. The set of indices for tokens generated by the model is denoted as $\mathcal{I}_{gen}$.

A.2 Architecture of LLMs

The input sequence is processed by the LLM through L transformer blocks (Vaswani et al., 2017). The update process of residual stream at each layer l can be unified as follows:

$$\hat{x}^l = \text{MHA}^l(\text{LN}(x^{l-1})) + x^{l-1} \quad (1)$$

$$x^l = \text{Trans}^l(\text{LN}(\hat{x}^l)) + \hat{x}^l \quad (2)$$

where $x^l \in R^{n \times d}$ represents the hidden state at layer l, d is the embedding dimension, and n is the total sequence length. MHA denotes the Multi-Head Attention module, and Trans represents the transition layer, which is a Feed-Forward Network (FFN) in dense LLMs or a Mixture-of-Experts (MoE) layer in MoE LLMs.

In the l-th layer, the MHA mechanism allocates attention weights by computing the similarity between Queries (Q) and Keys (K):

$$\alpha_{i,j}^{l,h} = \text{softmax}\left(\frac{(x_i^{l-1} W_Q^{l,h})(x_j^{l-1} W_K^{l,h})^T}{\sqrt{d_k}}\right) \quad (3)$$

where $\alpha_{i,j}^{l,h}$ denotes the attention weight from token i to token j in the h-th attention head of the l-th layer. The final output of a single attention head is the aggregation of contributions from all preceding tokens:

$$\text{MHA}^{l,h}(x_i) = \sum_{j \leq i} \alpha_{i,j}^{l,h} (x_j^{l-1} W_{OV}^{l,h}) \quad (4)$$

Here, $W_{OV}^{l,h} = W_V^{l,h} W_O^{l,h}$ represents the joint value-output projection matrix, which defines the specific

path through which information is mapped from input tokens into the residual stream.

For MoE LLMs, the transition layer introduces a sparse activation mechanism where a routing function G selects the Top-k experts for each token. The output is scaled by the routing weights and aggregated:

$$\mathrm{MoE}(x_i) = \sum_{e \in \text{Top-k}(G_i)} G_{i,e} \cdot E_e(\mathrm{LN}(x_i)) \quad (5)$$

where $G_i = \mathrm{softmax}(x_i W_G)$ and E_e denotes the e-th expert. Through this hierarchical process, the LLM transforms internal representations into the final response.

B Interpretability Analysis Methods

This section introduces our core methodology for revealing the internal mechanisms of LLMs during tabular reasoning, categorized into three dimensions: *Inner Prompt*, *Inner Table*, and *Contribution*.

B.1 Inner Prompt: Segment Attention Ratio

To quantify the allocation of attention resources across different input segments during inference, we define the **segment attention ratio** $\mathcal{D}$, which measures the degree of attention weight assigned to a specific segment $\mathcal{S}$ at layer l, averaged across all attention heads and the entire generation sequence $\mathcal{I}_{gen}$:

$$\mathcal{D}^l(\mathcal{S}) = \frac{1}{|\mathcal{I}_{gen}| \cdot H} \sum_{i \in \mathcal{I}_{gen}} \sum_{h=1}^{H} \sum_{j \in \mathcal{S}} \hat{\alpha}_{i,j}^{l,h} \quad (6)$$

where $\hat{\alpha}_{i,j}^{l,h} = \frac{\alpha_{i,j}^{l,h}}{\sum_{k \in \mathcal{I}_{in}} \alpha_{i,k}^{l,h}}$ represents the **rescaled weights for the input segment $\mathcal{I}_{in}$ at generation step** $i \in \mathcal{I}_{gen}$. By observing the trend of $\mathcal{D}^l(\mathcal{S})$ across layers, we can identify the dynamic transition of attention from the structured background ($\mathcal{S}_{tab}$) to the task objective ($\mathcal{S}_q$).

B.2 Inner Table: Table Attention Entropy

To qualitatively describe the degree of focus on relevant cells within the table, we define the **average table attention entropy** at layer l as:

$$\mathcal{H}_{table}^l = \frac{1}{|\mathcal{I}_{gen}|} \sum_{i \in \mathcal{I}_{gen}} \left(- \sum_{j \in \mathcal{I}_{tab}} \bar{P}_i^l(j) \log \bar{P}_i^l(j) \right) \quad (7)$$

where the inner-table attention distribution $\bar{P}_i^l(j)$ is defined as:

$$\bar{P}_i^l(j) = \frac{\sum_{h=1}^{H} \alpha_{i,j}^{l,h}}{\sum_{k \in \mathcal{I}_{tbl}} \sum_{h=1}^{H} \alpha_{i,k}^{l,h}}, \quad \forall j \in \mathcal{I}_{tab} \quad (8)$$

A higher entropy indicates a uniform distribution of attention across the table, whereas a lower entropy signifies that attention is concentrated in a few specific cells. This implies that the layer may identify the task-relevant information.

B.3 Value-Weighted Contribution

Following the perspective of previous researches (Kobayashi et al., 2020; Gu et al., 2024; Kang et al., 2025a) that attention weights alone do not quantify the actual magnitude of change in the residual stream, we introduce the **Value-Weighted Contribution** analysis. To capture the model's behavior throughout the generation process, we compute the averaged contribution over $\mathcal{I}_{gen}$:

$$C^l(\mathcal{S}) = \frac{1}{|\mathcal{I}_{gen}| \cdot H} \sum_{i \in \mathcal{I}_{gen}} \left\| \sum_{j \in \mathcal{S}} \sum_{h=1}^{H} \alpha_{i,j}^{l,h} x_j^{l-1} W_{OV}^{l,h} \right\|_2 \quad (9)$$

where $\mathcal{S} \in \{\mathcal{I}_{sys}, \mathcal{I}_{tab}, \mathcal{I}_q\}$. By integrating the hidden states x_j with the projection matrix W_{OV}, this formula faithfully reconstructs the intensity of the information volume injected by a specific segment $\mathcal{S}$ at layer l to drive the following reasoning.

C Experimental Details

C.1 Prompt Templates

We provide the complete prompt templates used in our experiments. The input structure follows the standard practice of prompting LLMs for table tasks, consisting of three segments: (1) a prompt template, (2) a table content segment, and (3) a user question segment.

Overall Prompt Template

System Prompt:

```
You are a helpful assistant.
```

User Prompt:

```
TABLE:
{table_content}
INPUT:
{question_content}
```

Prompt Template for Table Question Answering (WTQ, HiTab, AITQA)

TABLE:
{table_content (markdown / html)}

INPUT:
Given the table titled '{table_title}', answer the following question based on the given table. The final answer should be concise and use the following format:

```json
{
    "answer": ["answer1", "answer2", ...]
}
```

Question: {question_content}

Prompt Template for Table Fact Verification (TabFact)

TABLE:
{table_content (markdown / html)}

INPUT:
Given the table titled '{table_title}', determine whether the following statement is entailed or refuted by the given table (Output 1 for entailed and 0 for refuted):
Statement: {statement_content}

C.2 Table Serialization Formats

Unless otherwise specified, we use Markdown as the default table serialization format. Figure 7 illustrates the two serialization formats (Markdown and HTML) used in our experiments.

Markdown Format

```
| Name  | Age | City    |
|-------|-----|---------|
| Alice | 25  | Beijing |
| Bob   | 30  | Shanghai|
```

HTML Format

```
<table>
  <tr><th>Name</th><th>Age</th>
      <th>City</th></tr>
  <tr><td>Alice</td><td>25</td>
      <td>Beijing</td></tr>
  <tr><td>Bob</td><td>30</td>
      <td>Shanghai</td></tr>
</table>
```

Figure 7: Examples of table serialization formats.

C.3 Evaluation Datasets Statistics

As summarized in Table 3, the benchmarks we select encompass both *flat tables* with simple row/column headers (WTQ, TabFact) and *hierarchical tables* with merged cells (HiTab, AITQA).

Table 3: Summary of 2,000 evaluation samples used in our empirical study.

Dataset	Task	Structure	Avg Tokens	#Samples
WTQ	TQA	Flat	1012	500
HiTab	TQA	Hierarchical	964	500
AITQA	TQA	Hierarchical	651	500
TabFact	TFV	Flat	681	500

This selection enables our analysis data to comprehensively cover table structures and tabular tasks encountered in real-world scenarios.

C.4 Analyzed Models

To provide a comprehensive analysis across different model architectures and training paradigms, we categorize the investigated LLMs into three groups:

(1) Instruct LLMs. We analyze general-purpose instruction-tuned models that represent the mainstream LLM paradigm, including the Llama series (Dubey et al., 2024) (e.g., Llama-3.1-8B-Instruct) and the Qwen series (Yang et al., 2024) (e.g., Qwen-2.5-7B-Instruct).

(2) Table-Specific Fine-tuned LLMs. To understand how specialized training affects internal mechanisms, we include models specifically optimized for tabular tasks. We further distinguish between *SFT-only* models trained through supervised fine-tuning (e.g., the TAMA series (Deng and Mihalcea, 2025)) and *RL-only* models optimized via reinforcement learning (e.g., the Table-R1-Zero series (Yang et al., 2025b)).

(3) Mixture-of-Experts (MoE) LLMs. Beyond dense architectures, we extend our analysis to MoE models to examine whether sparse expert activation leads to different table understanding behaviors. Representative models include OLMoE-1B-7B (Muennighoff et al., 2024), DeepSeek-V2-Lite-Chat (DeepSeek-AI et al., 2024), and Qwen3-30B-A3B (Yang et al., 2025a).

C.5 Artifact Use and Implementation Details

License of Artifacts. All benchmarks and analyzed LLMs are publicly released open-source artifacts. Our use is restricted to non-commercial academic research and is fully consistent with the license terms specified on their official release pages.

Intended Use of Artifacts. All datasets and models are used in a manner consistent with their original intended purposes.

Model	WTQ	TabFact	HiTab	AITQA
Instruct LLMs				
LLaMA3.2-1B-Instruct	6.00	5.40	6.40	9.80
LLaMA3.2-3B-Instruct	33.00	58.00	30.80	52.00
LLaMA3.1-8B-Instruct	50.20	70.80	49.00	57.80
Qwen2.5-3B-Instruct	52.40	78.00	49.80	55.60
Qwen3-4B-Instruct-2507	52.00	75.20	49.60	60.00
Qwen2.5-7B-Instruct	51.00	73.00	53.40	71.20
Qwen2.5-14B-Instruct	56.40	**88.60**	55.80	74.40
Qwen2.5-32B-Instruct	**64.60**	85.80	68.80	83.60
Table-Specific Fine-tuned LLMs				
TAMA-Qwen2.5	52.40	74.80	56.80	71.60
TAMA	50.60	72.60	66.80	82.00
TableGPT2	56.80	74.80	59.80	84.20
Table-R1-7B-Zero	53.80	76.20	63.80	**85.20**
Table-R1-8B-Zero	60.40	70.40	**74.00**	78.60
MoE LLMs				
OLMoE-1B-7B-Instruct	6.82	2.20	5.01	14.60
DeepSeek-V2-Lite-Chat	28.60	63.40	30.40	53.00
Qwen3-30B-A3B-Instruct-2507	56.40	78.20	57.40	73.20

Table 4: LLMs' performance (accuracy %) on evaluation data of four benchmarks. Best results are bolded.

Packages and Versions. Our implementation uses the latest stable releases of the major dependencies at the time of experimentation, including PyTorch, HuggingFace Transformers, and vLLM for model inference, with NumPy, Matplotlib, and scikit-learn for analysis. For each LLM, we directly load the official HuggingFace checkpoint with its default tokenizer and chat template, without any model-side hyperparameter tuning.

D Model Analysis Details

D.1 Benchmark Performance

Table 4 reports the accuracy of all analyzed LLMs on our four evaluation benchmarks. Several observations emerge: (1) larger models generally achieve better performance; (2) table-specific fine-tuned models (TAMA, TableGPT2, Table-R1) show notable improvements on hierarchical table datasets (HiTab, AITQA); and (3) RL-tuned models (Table-R1 series) demonstrate particularly strong performance on complex table structures.

Table 6 reports the performance of representative models under different input format and inference strategy combinations, with 95% bootstrap confidence intervals (1,000 iterations, N=2,000 per condition). All CoT vs. Direct differences are statistically significant (non-overlapping CIs), confirming the reliability of prompting strategy conclusions. Markdown vs. HTML differences fall within the margin of uncertainty for most models, consistent with the observation that format-specific differences in early layers gradually converge in deeper layers, resulting in minimal overall performance gaps.

D.2 Analysis of Effective Depth

We extend the effective depth analysis from Section 5 to all analyzed models. Table 5 summarizes the layer effectiveness statistics, while Figure 27 provides detailed layer-wise prediction stability curves for each model.

As described in the main text, we define the *effective layer* as the first layer where the top-5 token overlap between the intermediate prediction and the final output exceeds 0.3. The *effective ratio* (Eff. Ratio = Eff. Layers / Total Layers) indicates how deeply the model processes before reaching a stable prediction; a lower ratio suggests earlier stabilization.

Key observations from the extended analysis:

- Most models stabilize their predictions in the

Model	Total Layers	Eff. Layers	Eff. Ratio
Instruct LLMs			
Llama-3.2-1B-Instruct	16	14	0.88
Llama-3.2-3B-Instruct	28	23	0.82
Meta-Llama-3.1-8B-Instruct	32	25	0.78
Qwen2.5-3B-Instruct	36	33	0.92
Qwen2.5-7B-Instruct	28	25	0.89
Qwen2.5-14B-Instruct	48	44	0.92
Qwen2.5-32B-Instruct	64	57	0.89
Qwen3-4B-Instruct-2507	36	24	0.67
Table-Specific Fine-tuned LLMs			
TableGPT2-7B	28	25	0.89
TAMA-QWen2.5	28	25	0.89
Table-R1-Zero-7B	28	25	0.89
TAMA-1e-6	32	25	0.78
Table-R1-Zero-8B	32	24	0.75
MoE LLMs			
OLMoE-1B-7B-0924-Instruct	16	14	0.88
DeepSeek-V2-Lite-Chat	27	21	0.78
Qwen3-30B-A3B-Instruct-2507	48	45	0.94

Table 5: Layer effectiveness analysis on table-related tasks using LogitLens. Eff. Layers identifies the first layer where the predicted vocabulary distribution aligns with the final output (defined by the top-5 token overlap first exceeds 0.3). Eff. Ratio (Eff. Layers / Total Layers) reflects the stage at which the model stabilizes its prediction; a lower ratio indicates that the model reaches its final decision in its shallower layers.

Table 6: Performance (%) with 95% bootstrap confidence intervals under different input formats and inference strategies. † marks statistically significant differences between CoT and Direct (non-overlapping CIs).

	Markdown		HTML	
Model	Direct	CoT	Direct	CoT
Instruct LLMs				
Qwen2.5-7B	62.2 ±2.1	67.2 ±2.0†	61.4 ±2.1	67.8 ±2.0†
Llama-3.1-8B	57.0 ±2.2	62.4 ±2.1†	58.0 ±2.1	58.4 ±2.1†
Table-Specific Fine-tuned LLMs				
Table-R1-7B	69.8 ±2.0	82.3 ±1.7†	70.3 ±1.9	81.6 ±1.7†
Table-R1-8B	70.8 ±2.0	82.2 ±1.7†	69.9 ±2.1	80.6 ±1.8†
MoE LLMs				
Qwen3-30B-A3B	66.3 ±2.1	70.2 ±2.0†	68.3 ±2.0	72.1 ±1.9†

later 75–90% of their total layers, confirming that table tasks require deep processing.

- Table-specific fine-tuning (TAMA, Table-R1) does not significantly alter the effective depth compared to their base models, suggesting that layer-wise functionality is largely determined during pre-training.
- MoE models (OLMoE, DeepSeek-V2-Lite, Qwen3-30B-A3B) exhibit similar effective ratios to dense models of comparable capacity.

E More Discussions

E.1 Connection to Retrieval Heads

Recent work has identified *retrieval heads* which is a sparse set of attention heads responsible for copying relevant information from context to output (Wu et al., 2024; Xiao et al.). Given that table QA inherently requires locating and extracting specific cell values, we hypothesize that table understanding may share similar mechanisms.

To investigate this, we compare the attention heads most critical for table tasks (measured by attention norm intensity) with the Top-32 retrieval heads identified by Wu et al. (2024) using our analysis tabular data. As shown in Figure 9, the retrieval heads (marked with red boxes) predominantly align with regions of high attention norm intensity (yellow areas) across all examined models. This overlap is particularly pronounced in the

TABLE:
Rank	Nation	Gold	Silver	Bronze	Total
1	China	13	9	13	35
2	Japan	7	10	7	24
3	Uzbekistan	1	2	3	6
4	Kazakhstan	2	2	0	4
5	North Korea	1	0	1	2
6	South Korea	0	0	2	2
Total	Total	24	23	26	73

INPUT:
Given the table titled 'Figure skating at the Asian Winter Games', Answer the following question based on the given table. The final answer should be concise and use the following format:
```json
{
    "answer": [ "answer1", "answer2", ... ]
}
```
Question: which nation won the most gold medals? **Prompt**

Qwen2.5-7B-Instruct: ```json\n{\n \"answer\": [\n \"China\"\n]\n}\n```
Meta-LLaMA-3.1-8B-Instruct: {\n \"answer\": [\n \"China\"\n]\n} **Response**

Figure 8: The qualitative case example for Q1.

middle-to-late layers: for instance, in Qwen2.5-7B, retrieval heads cluster around layers 19–27; in LLaMA3.1-8B, they concentrate in layers 14–15 and 26–28; in TAMA and Table-R1-8B, similar patterns emerge in layers 13–17 and 21–28.

Notably, this alignment pattern persists across both vanilla instruction-tuned models (Qwen2.5, LLaMA3) and table-specific fine-tuned models (TAMA, Table-R1), suggesting that table-specific training does not fundamentally alter which heads are responsible for information retrieval—it may instead enhance the efficiency of these pre-existing circuits.

This finding suggests that LLMs leverage their pre-existing retrieval mechanisms for table understanding, rather than developing entirely separate circuits. It also has practical implications: KV cache compression techniques designed for retrieval heads (Xiao et al.) may be directly applicable to accelerate table reasoning tasks without significant performance degradation.

E.2 Attention-answer correspondence

To complement the causal analysis, we select 200 lookup-type examples and identify the Top-20 attention-weighted cells in low-entropy layers (layers 10–15) for Llama-3.1-8B and Qwen2.5-7B. Cells focused on by low-entropy layers directly correspond to the model's final response in 53.0% (Llama) and 43.9% (Qwen) of cases. When low-entropy layers successfully attend to the correct answer cell, final prediction accuracy reaches 65.6% (Llama) and 71.2% (Qwen). Conversely, when models answer incorrectly, 52.8% (Llama) and 74.4% (Qwen) of errors are associated with low-entropy layers failing to attend to the correct cell. This suggests that misfocused middle-layer attention is a frequent predictor of incorrect predictions, pointing to potential utility for attention-based error detection.

F Related Work

F.1 LLM-based Table Understanding

Table understanding (TU) technique aims to enable models to automatically comprehend tables and perform various tabular tasks based on user requests (Shigarov, 2023; Lu et al., 2025). With the rapid iteration of LLMs, their ability has opened new possibilities for more intelligent TU applications. One line of research endeavours to enhance LLMs' table understanding ability through different strategies. For instance, eliciting correct reasoning steps with prompt engineering and carefully selected in-context examples (Chen, 2023; Ye et al., 2023; Wang et al., 2024b; Ji et al., 2024; Jiang et al.,

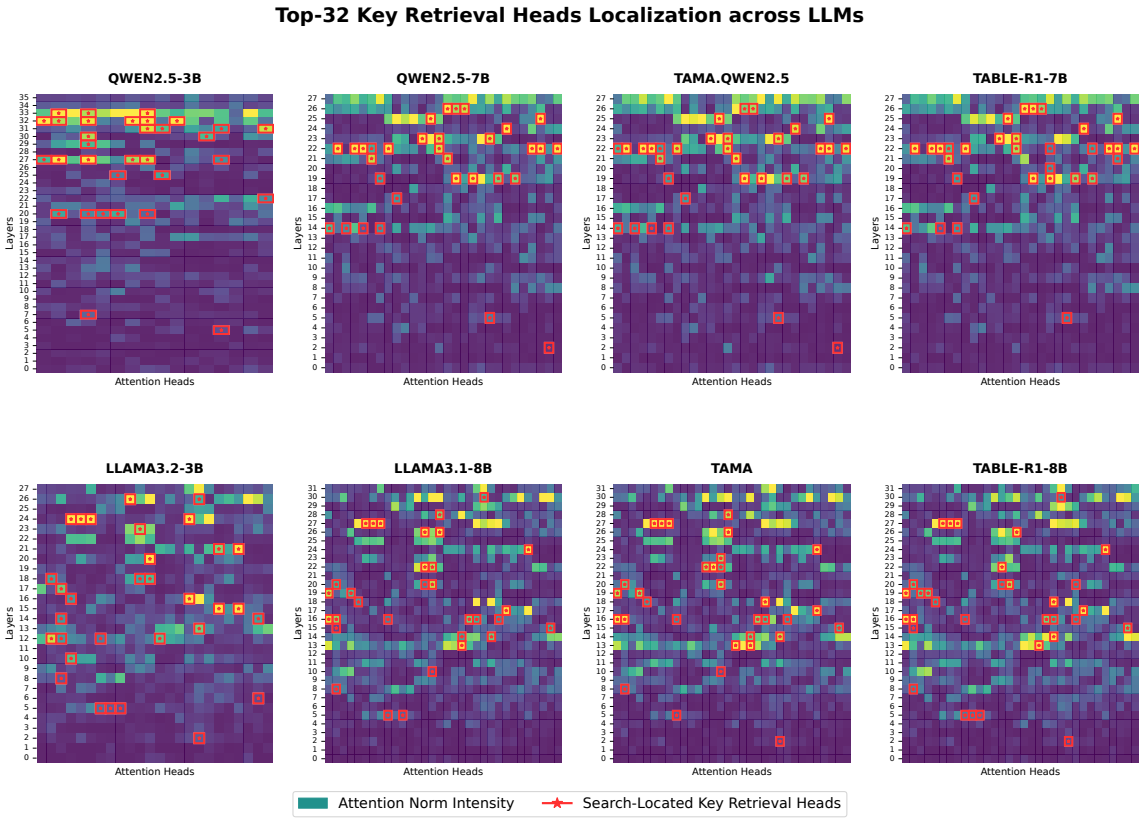

Figure 9: Top-32 retrieval heads' overlap with attention norm heatmaps across LLMs. Background color indicates attention norm intensity (yellow = higher); red boxes mark the Top-32 retrieval heads identified by prior work. The substantial overlap, especially in middle-to-late layers, suggests that table understanding reuses pre-existing retrieval circuits.

2023), collecting table instruction tuning data for supervised fine-tuning (Zhang et al., 2024, 2025), exploring new tasks and reward strategies for reinforcement learning (Yang et al., 2025b; Wu et al., 2025; Zhou et al., 2025a), and building powerful table agents for multi-turn data analysis and excel manipulation (Li et al., 2023; Weng et al., 2025; Wang et al., 2025).

In addition to the performance-oriented studies mentioned above, another valuable direction is to explore the robustness and interpretability of LLMs' table-related capabilities. For example, evaluating models' performance against divergent perturbations such as table formats (Sui et al., 2024), noisy operators (Singha et al., 2023) and cell replacement (Bhandari et al., 2025). Compared with existing work that mainly focused on the performance robustness, we aim at seeking a better understanding of the underlying mechanisms of LLM-based table understanding and provide valuable insights for future follow-ups.

F.2 Interpretability and Analysis of LLMs

Despite the strong capabilities of LLMs, understanding their internal mechanisms remains a key challenge. Some studies focus on the roles of different layers in LLMs, examining how individual layers contribute to information processing (Skean et al., 2025) and how semantic representations are gradually constructed across layers (Csordás et al., 2025; Hu et al., 2025). Other lines of work investigate specific internal components, such as attention heads in self-attention mechanisms and feed-forward network (FFN) modules, with the aim of clarifying their roles in knowledge representation (Geva et al., 2021; Meng et al., 2022) and information acquisition (Wu et al., 2024; Kobayashi et al., 2023), revealing various intriguing phenomena within LLMs, such as attention sink (Kobayashi et al., 2020; Xiao et al., 2023; Gu et al., 2024; Qiu et al., 2025; Queipo-de Llano et al., 2025). Furthermore, with the growing adoption of Mixture-of-Experts (MoE) architectures, recent researches (Wang et al., 2024a; Bai et al., 2025; Lo et al., 2025; Bandarkar et al., 2025; Su et al., 2025)

analyze expert modules within MoE models, exploring their specialization patterns and activation behaviors during task execution.

While these efforts have provided valuable insights (Si et al., 2023; Bavaresco et al., 2025), they predominantly focus on unstructured text tasks such as math reasoning and factual recall. In this work, we extend the scope of interpretability research to structured table data, conducting empirical analyses to explore how LLMs process tables internally and offering preliminary findings across several dimensions.

G Attention Dynamics for All Models

To validate the generalizability of our findings in Section 4, we present the complete attention dynamics analysis for all 16 LLMs examined in this study. Each figure follows the same layout as Figure 1: upper panels show case-level attention patterns at selected layers, while lower panels display aggregated metrics including segment attention ratio, table attention entropy, and attention contribution across all layers. Despite variations in model architecture (dense vs. MoE), scale (1B to 32B), and training paradigm (instruction-tuning, SFT, RL), all models exhibit the consistent three-phase pattern: dispersed attention in early layers, concentrated focus on query-relevant cells in middle layers, and amplified contributions in late layers. Notably, table-specific fine-tuned models (TAMA, Table-R1, TableGPT2) show slightly higher attention ratios to table content compared to their base models, while MoE models (DeepSeek-V2-Lite, OLMoE, Qwen3-30B-A3B) display more gradual entropy transitions across layers.

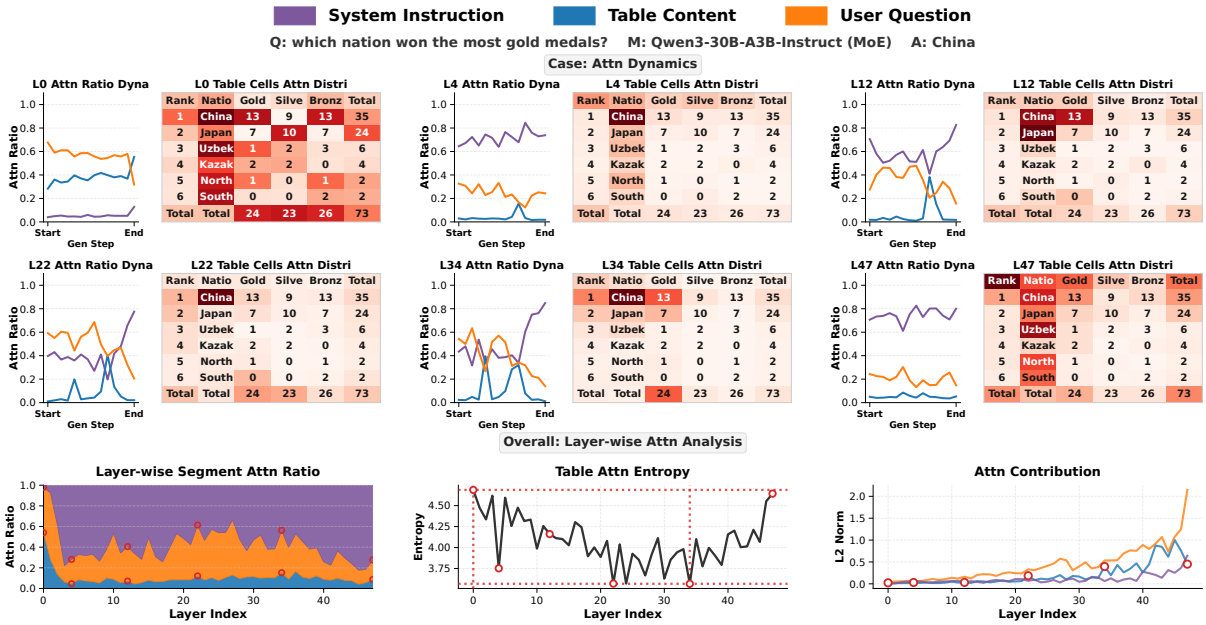

Figure 10: Attention dynamics of Qwen3-30B-A3B-Instruct-2507.

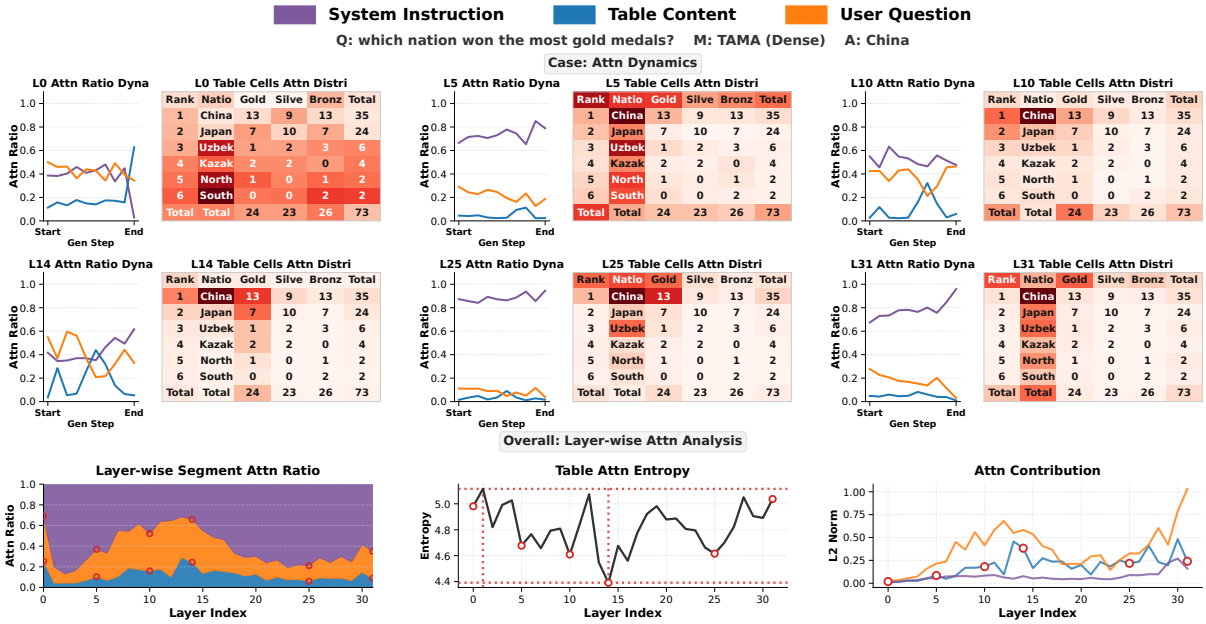

Figure 11: Attention dynamics of TAMA.

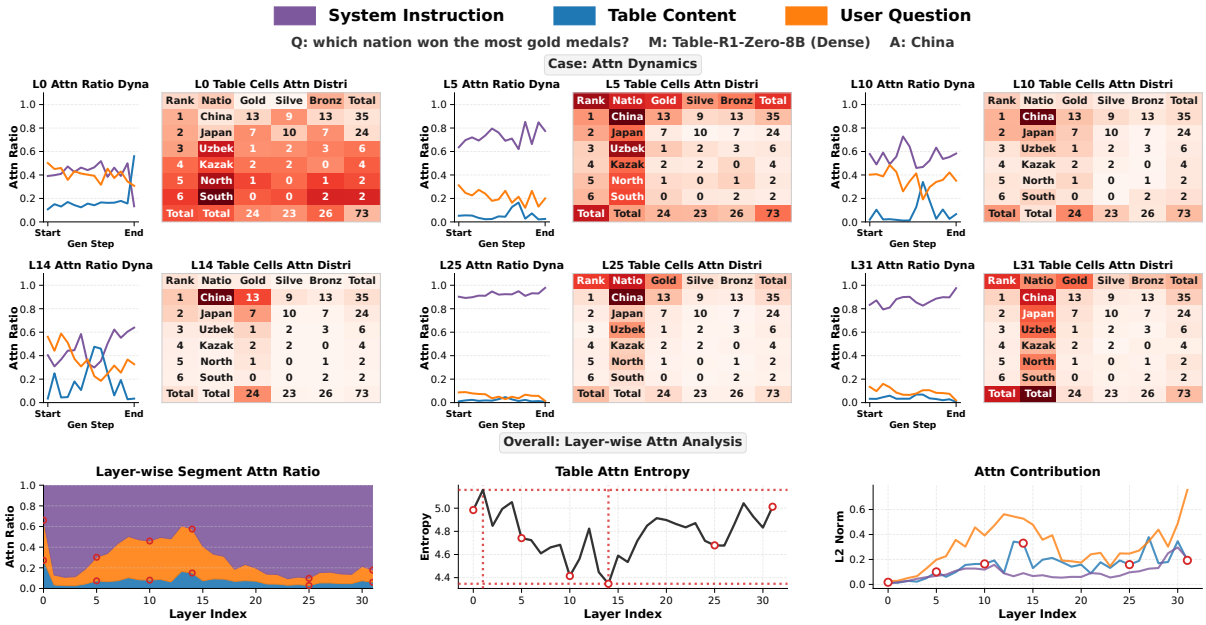

Figure 12: Attention dynamics of Table-R1-Zero-8B.

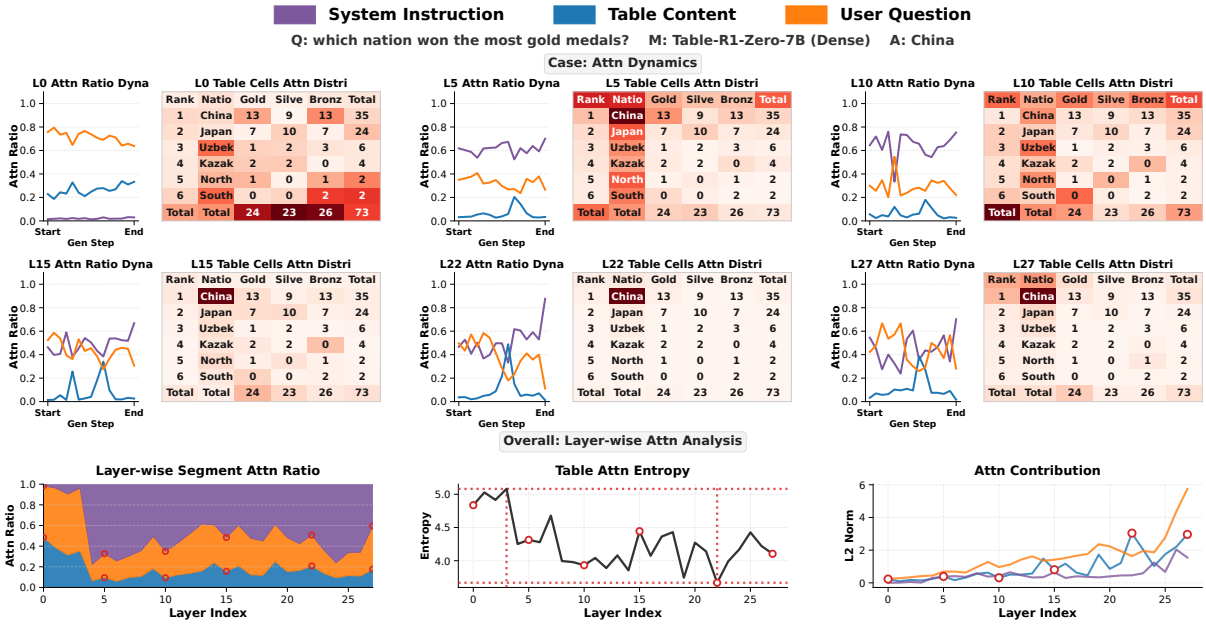

Figure 13: Attention dynamics of Table-R1-Zero-7B.

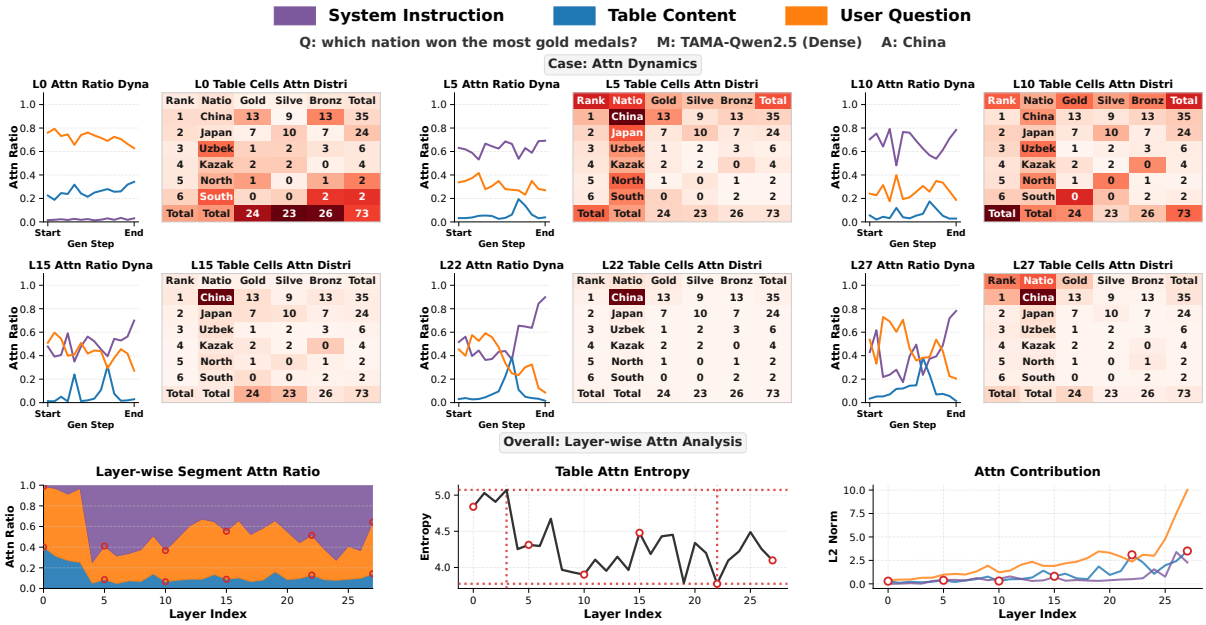

Figure 14: Attention dynamics of TAMA-Qwen2.5.

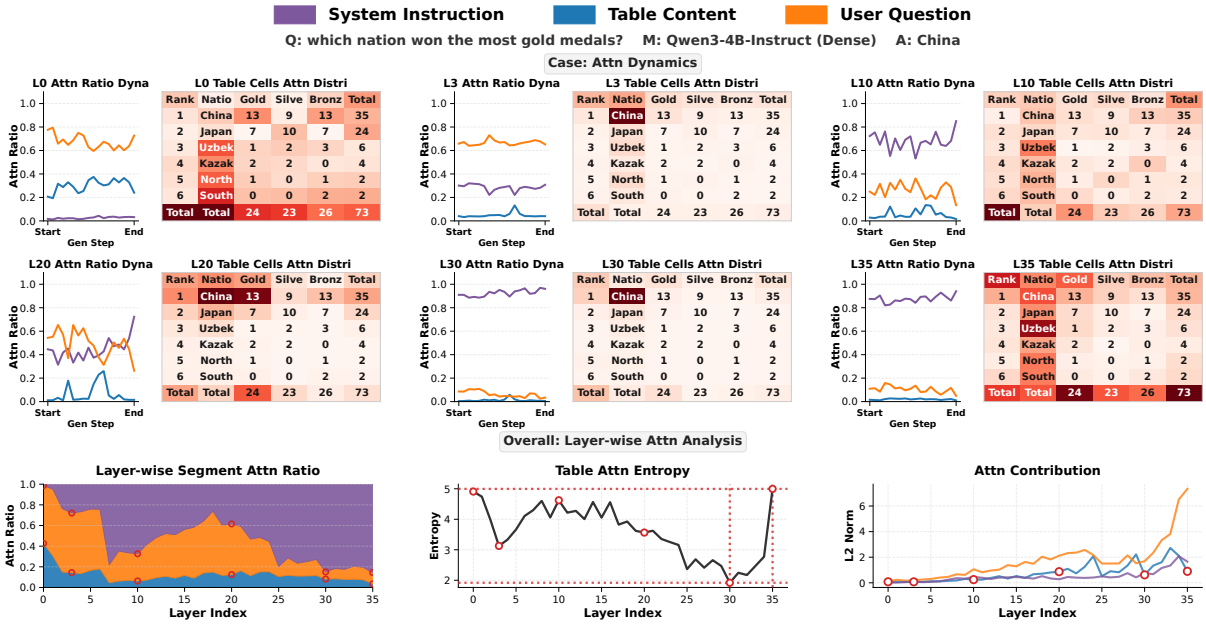

Figure 15: Attention dynamics of Qwen3-4B-Instruct.

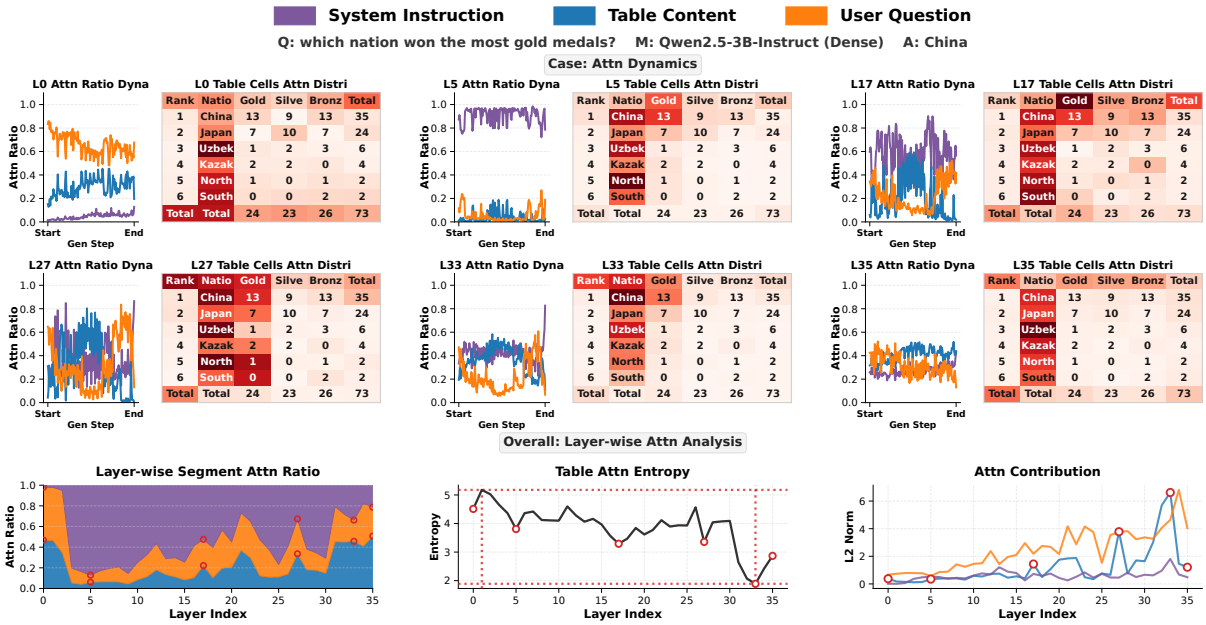

Figure 16: Attention dynamics of Qwen2.5-3B-Instruct.

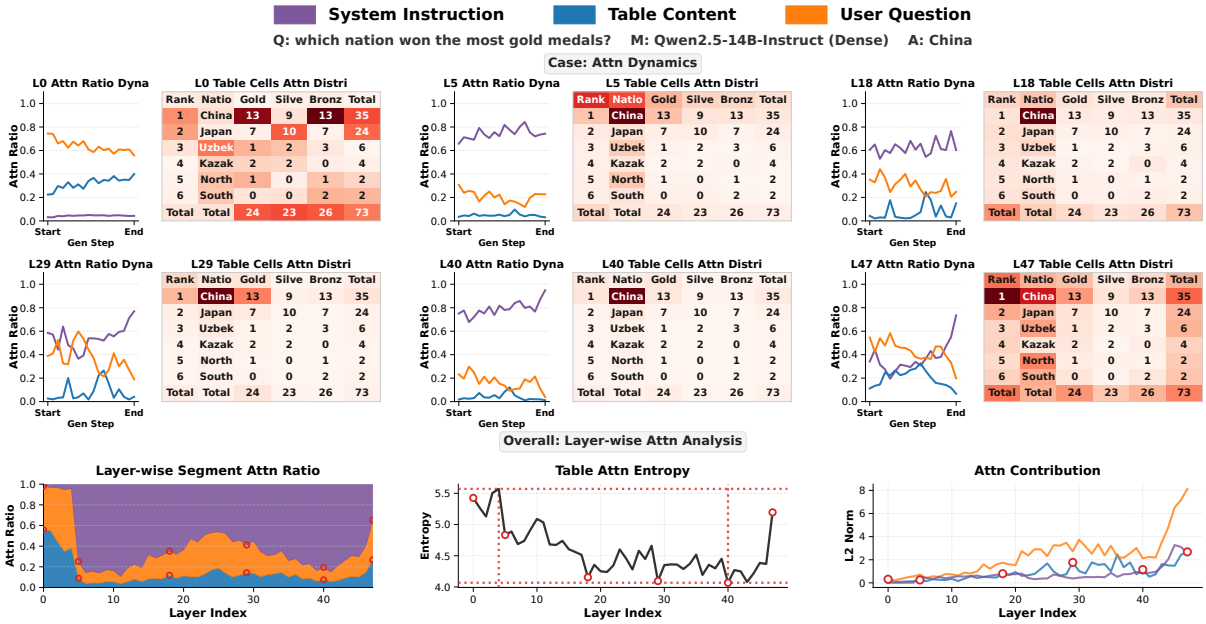

Figure 17: Attention dynamics of Qwen2.5-14B-Instruct.

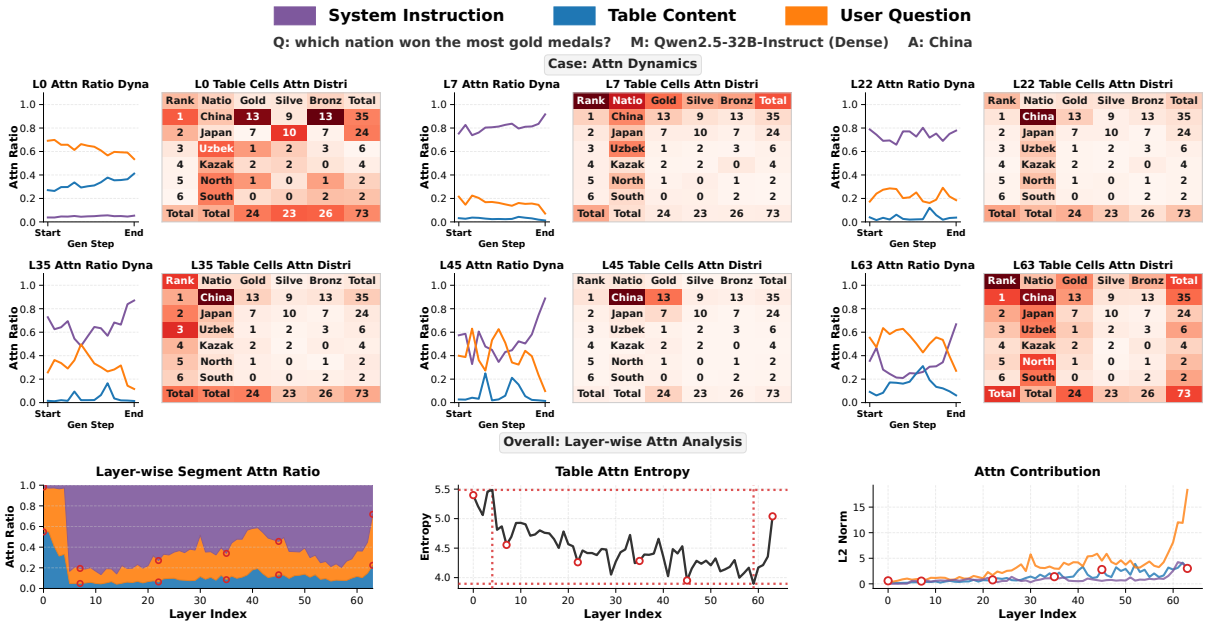

Figure 18: Attention dynamics of Qwen2.5-32B-Instruct.

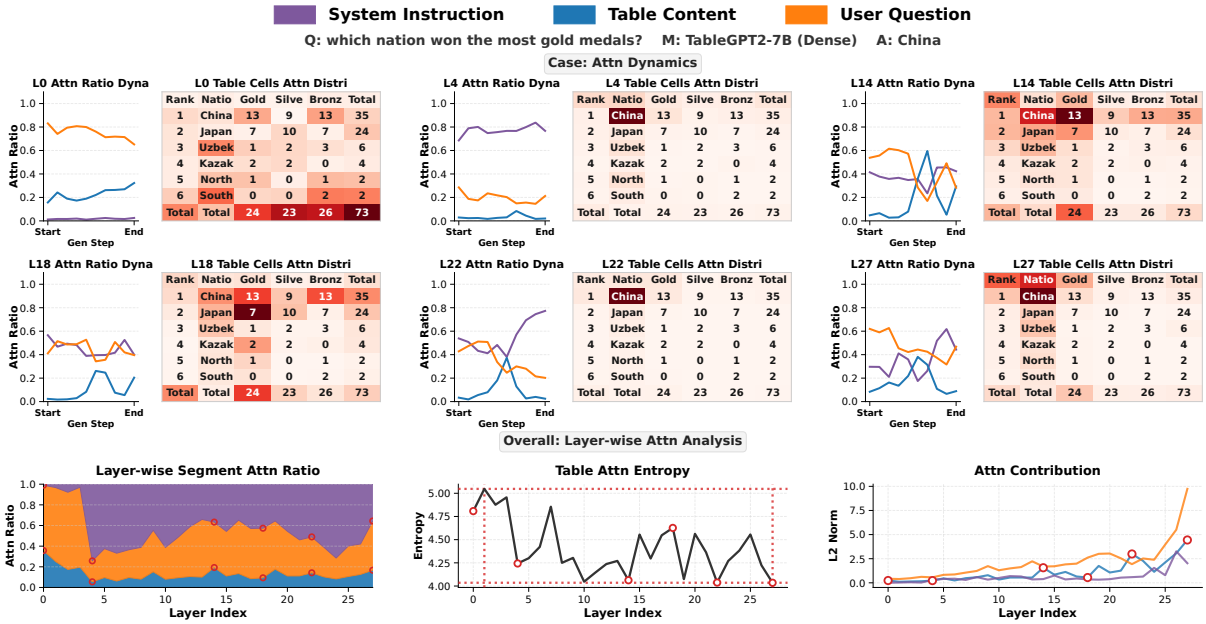

Figure 19: Attention dynamics of TableGPT2-7B.

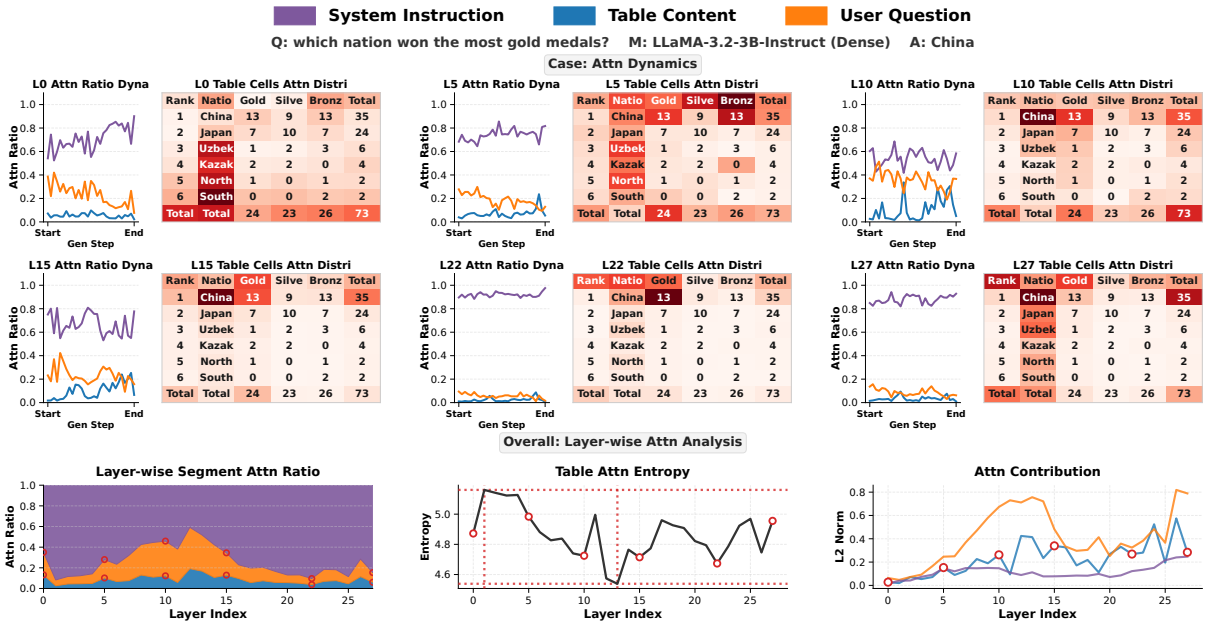

Figure 20: Attention dynamics of LLaMA3.2-3B-Instruct.

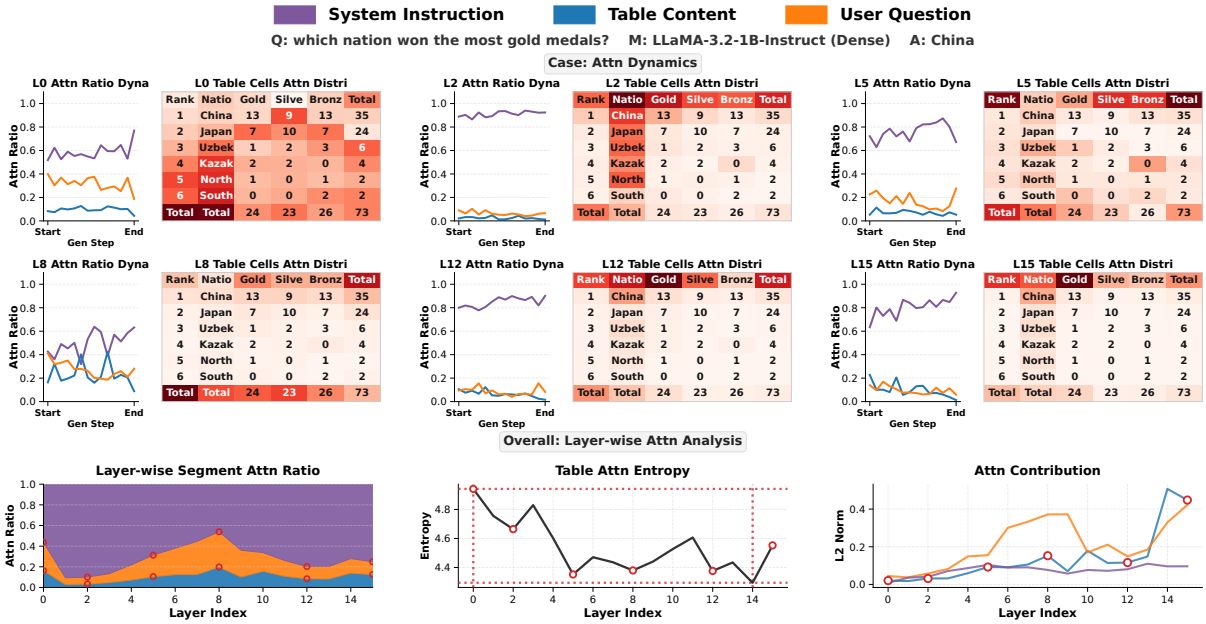

Figure 21: Attention dynamics of LLaMA3.2-1B-Instruct.

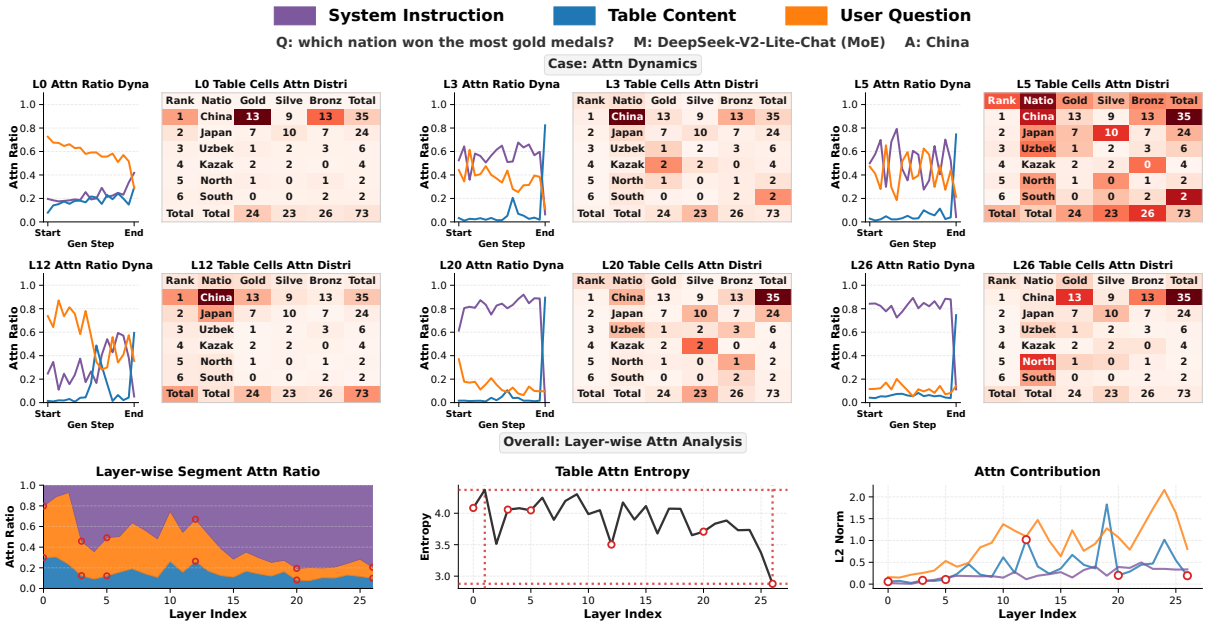

Figure 22: Attention dynamics of DeepSeek-V2-Lite-Chat.

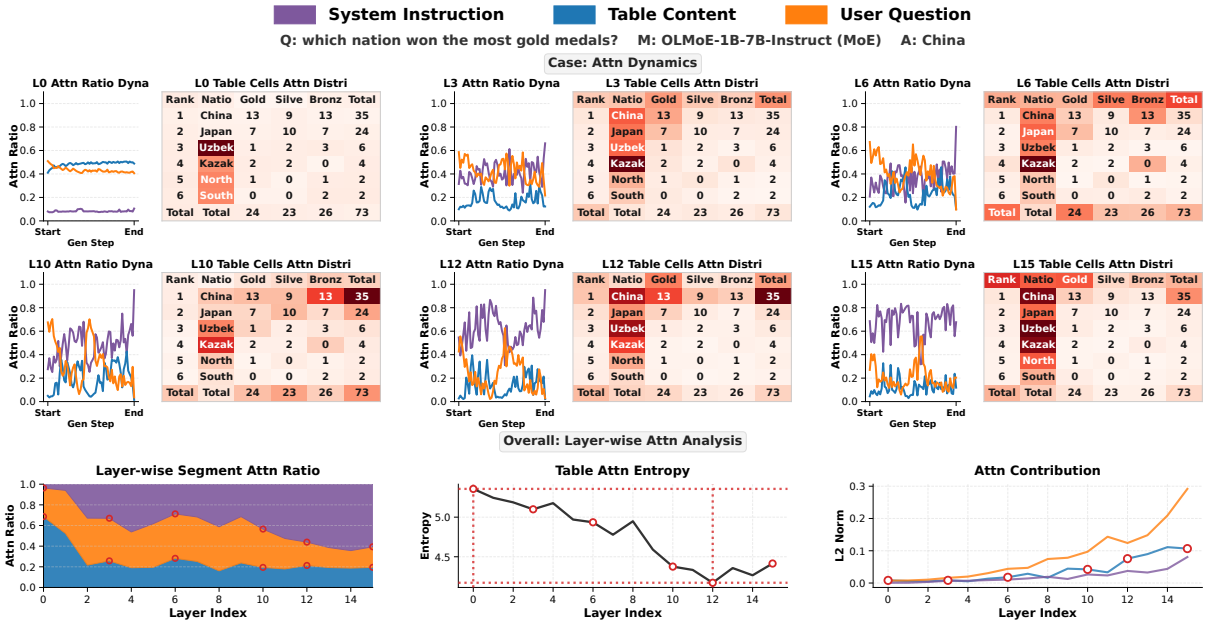

Figure 23: Attention dynamics of OLMoE-1B-7B-Instruct.

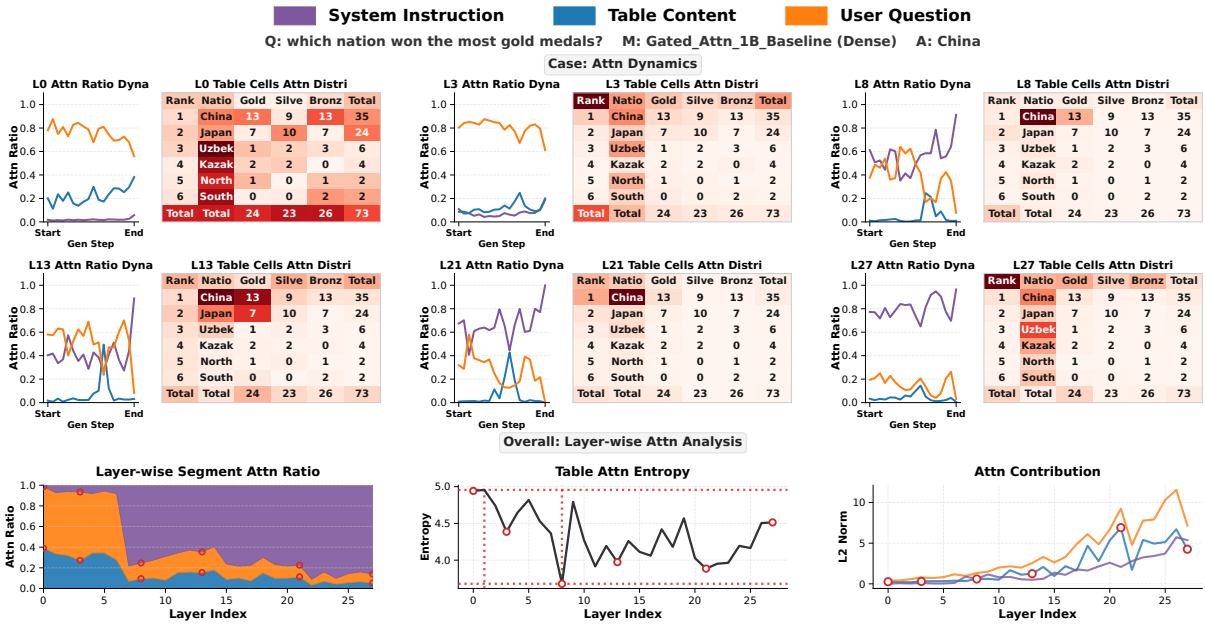

Figure 24: Attention dynamics of Qwen's 1B gated attention baseline.

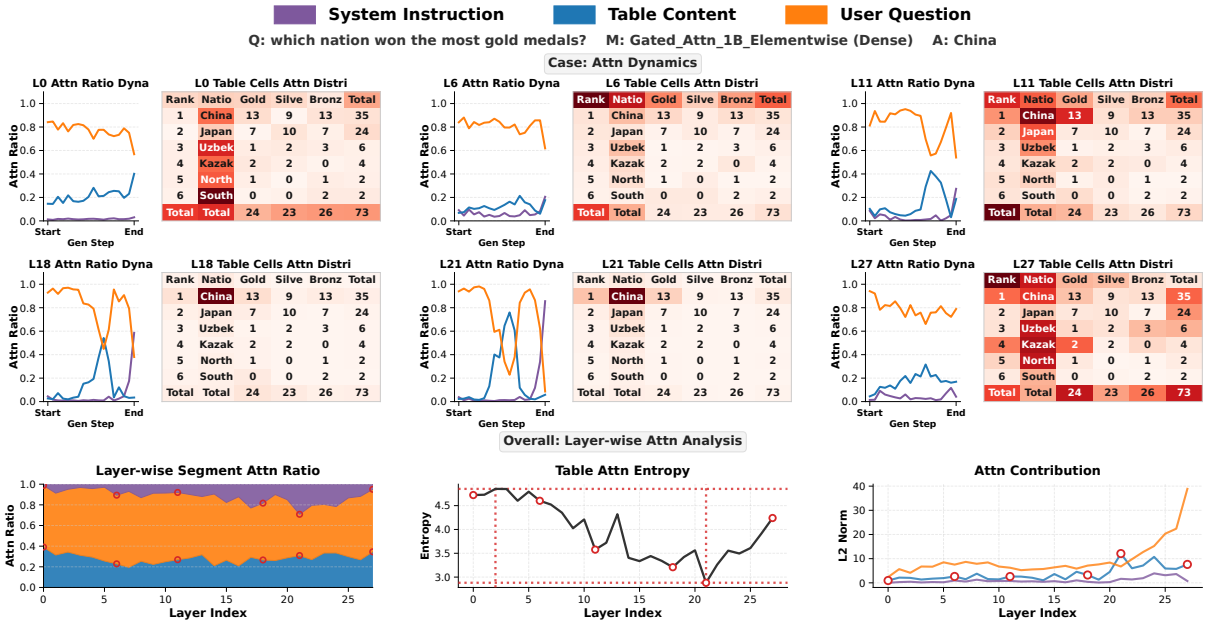

Figure 25: Attention dynamics of Qwen's 1B model with element-wise gated attention.

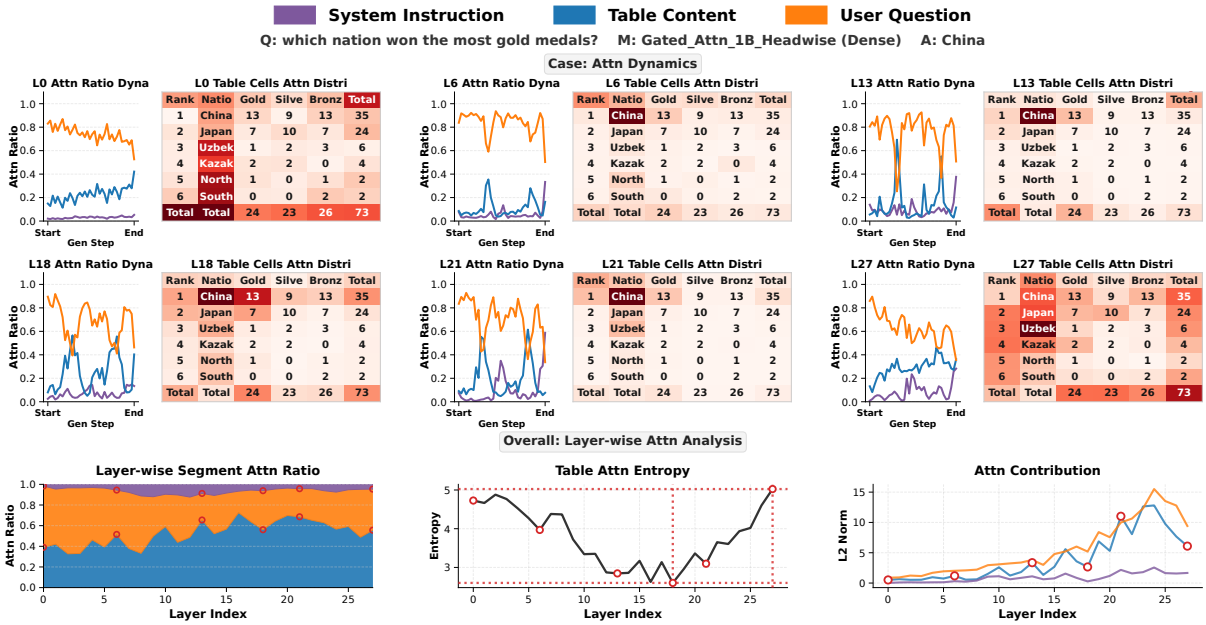

Figure 26: Attention dynamics of Qwen's 1B model with Headwise Gated Attention.

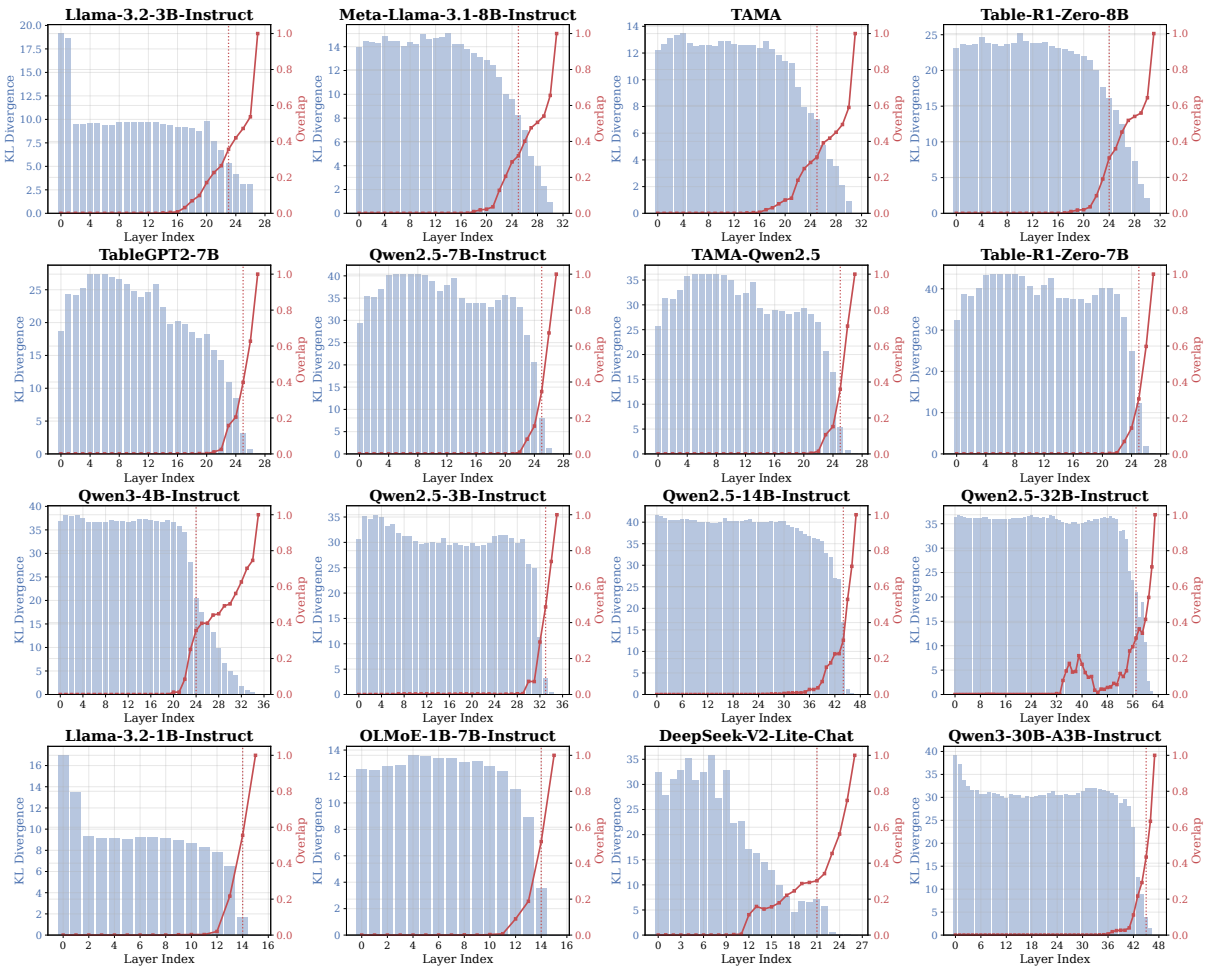

Figure 27: Analysis results of LLMs' effectiveness layer.