TabFlash: Efficient Table Understanding with Progressive Question Conditioning and Token Focusing

Jongha Kim[1], **Minseong Bae**[2*], **Sanghyeok Lee**[2*], **Jinsung Yoon**[3], **Hyunwoo J. Kim**[2†]
[1]Korea University
[2]KAIST
[3]Google Cloud AI
jonghakim@korea.ac.kr
{bms2002,sanghyeoklee,hyunwoojkim}@kaist.ac.kr
jinsungyoon@google.com

Abstract

Table images present unique challenges for effective and efficient understanding due to the need for question-specific focus and the presence of redundant background regions. Existing Multimodal Large Language Model (MLLM) approaches often overlook these characteristics, resulting in uninformative and redundant visual representations. To address these issues, we aim to generate visual features that are both informative and compact to improve table understanding. We first propose progressive question conditioning, which injects the question into Vision Transformer layers with gradually increasing frequency, considering each layer's capacity to handle additional information, to generate question-aware visual features. To reduce redundancy, we introduce a pruning strategy that discards background tokens, thereby improving efficiency. To mitigate information loss from pruning, we further propose token focusing, a training strategy that encourages the model to concentrate essential information in the retained tokens. By combining these approaches, we present TabFlash, an efficient and effective MLLM for table understanding. TabFlash achieves state-of-the-art performance, outperforming both open-source and proprietary MLLMs, while requiring 27% less FLOPs and 30% less memory usage compared to the second-best MLLM.

Code — https://github.com/mlvlab/TabFlash

1 Introduction

Table data is a vital information source, widely used to organize and communicate structured knowledge across diverse domains. With the recent success of Multimodal Large Language Models (MLLMs) (Alayrac et al. 2022; Li et al. 2023; Liu et al. 2023; Chen et al. 2024b), MLLM-based methods have gained popularity for table image understanding (Zheng et al. 2024; Zhao et al. 2024; Zhou et al. 2025). While these methods demonstrate the potential of MLLMs in table understanding, they often overlook the unique challenges posed by table images.

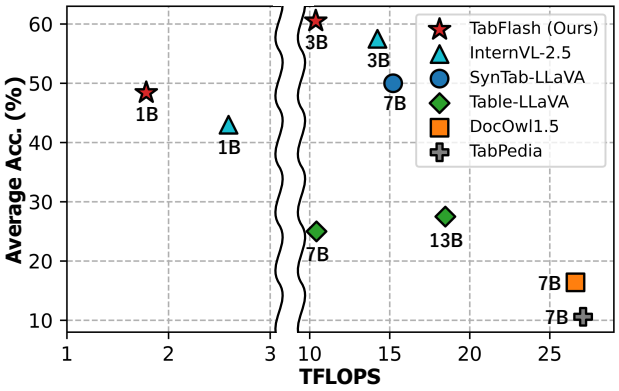

Figure 1: **Performance-cost comparison.** TFLOPs (x-axis) and average accuracy on 7 benchmarks (y-axis) are plotted. We propose TabFlash, an efficient MLLM with superior table understanding capability (Tab. 1) with significantly lower computational cost and GPU memory usage (Tab. 2).

Unlike natural images, table images require focused attention on localized regions relevant to a specific question, as the majority of the image content is typically irrelevant to the target task. In addition, they often include substantial redundancy, such as empty or background areas. Existing MLLMs struggle to handle these characteristics effectively, leading to the generation of uninformative and redundant visual representations. This not only degrades performance but also incurs high inference costs. To address these limitations, we aim to produce compact and informative visual representations tailored to the unique structure of table images.

As a solution, we first introduce progressive question conditioning, a strategy that injects question information into the Vision Transformer (ViT) (Dosovitskiy et al. 2021) to produce more informative visual features. Specifically, we embed the question into ViT layers with gradually increasing conditioning frequencies. In this design, early layers are conditioned less frequently, while later layers are conditioned more often. This approach is based on the observation

*This work was conducted at Korea University.
†Corresponding Author.

that early layers are more sensitive and unstable, whereas later layers are more stable and better suited to incorporate additional information (Dosovitskiy et al. 2021; Raghu et al. 2021). By adjusting the conditioning frequency to match each layer's capacity, progressive conditioning enables stable and effective integration of question context. As a result, the ViT generates visual features that are more closely aligned with the question, improving their informativeness.

In addition, we propose a background pruning strategy to generate more compact visual features. Previous MLLM-based approaches (Chen et al. 2024b; Zhao et al. 2024; Zhou et al. 2025) often provide a large number of visual tokens to the language model (LM), in some cases exceeding 3,000 tokens. Given the quadratic complexity of an LM concerning input tokens, this leads to a significant computational burden. However, much of a table image consists of redundant background pixels. We observe that the L_2 norm of output tokens from the ViT can serve as an effective signal to identify background regions, with tokens having lower norms typically corresponding to background areas (Fig. 3). Based on this observation, we introduce a token pruning strategy that removes low-norm tokens and passes only the retained tokens to the language model, reducing computation.

Still, we observe that a simple token pruning leads to significant performance loss. Through observation, we identify that useful information for answering the question is still present in tokens to be discarded. As a result, pruning those tokens without additional guidance causes information loss (Sec. 5.2). To address this issue, we propose token focusing, a training strategy that encourages the model to concentrate important information within the tokens that will be retained. Token focusing guides the model to produce correct answers based on the retained tokens, while discouraging correct predictions when only using the tokens designated for removal. By doing so, the model is promoted to store important information only on tokens to be retained during inference, thereby minimizing information loss.

Combining the proposed methods, we introduce TabFlash, an efficient MLLM for table understanding. These components work together to produce visual features that are both informative and compact, resulting in substantial improvements in effectiveness and efficiency. As shown in Fig. 1, TabFlash achieves state-of-the-art performance while requiring significantly lower computational resources. Remarkably, TabFlash even surpasses proprietary models such as GPT-4o and Gemini 2.5 Pro, highlighting both the challenge of table image understanding and the strength of our approach. In sum, our contributions are threefold:

- We propose progressive question conditioning, injecting questions to ViT layers in a progressively increasing frequency, thus obtaining question-aware visual features.
- We prune visual tokens based on their L_2 norm for efficiency. We also introduce token focusing, a training strategy that enforces information concentration on tokens to be retained, minimizing the information loss by pruning.
- We introduce TabFlash, an efficient MLLM achieving state-of-the-art results. TabFlash outperforms the second-best open-source model by 3 points while requiring 27% less FLOPs and 30% less memory. It also surpasses proprietary models, such as GPT-4o and Gemini 2.5 Pro.

2 Related Works

Table understanding with MLLMs. Table understanding involves interpreting and reasoning over tabular data, including tasks like question answering (Pasupat and Liang 2015; Zhu et al. 2021; Lu et al. 2023), fact verification (Chen et al. 2020; Gupta et al. 2020; Akhtar, Cocarascu, and Simperl 2022), and text generation (Lebret, Grangier, and Auli 2016; Wiseman, Shieber, and Rush 2017; Cheng et al. 2021). Following the success of Multimodal Large Language Models (MLLMs) across domains (Alayrac et al. 2022; Li et al. 2023; Liu et al. 2023; Park et al. 2024, 2025), MLLM-based architectures (Zhao et al. 2024; Zhou et al. 2025; Chen et al. 2024b; Zheng et al. 2024) have shown potential in table understanding tasks. Previous works largely focused on data construction and still struggled to interpret table images. Inspired by studies that condition vision encoders on input instructions (Abramovich et al. 2024; Ganz et al. 2024), we enhance table understanding by generating question-aware features while simultaneously reducing output tokens, yielding a more informative and compact representation.

Efficient MLLM. MLLMs suffer from high computational costs, primarily due to the quadratic complexity of self-attention in Language Models (LMs) concerning input token length. To address this, recent works have explored applying token pruning or merging strategies (Bolya et al. 2023; Liang et al. 2022; Lee, Choi, and Kim 2024; Choi et al. 2024) for MLLMs. For instance, FastV (Chen et al. 2024a), FitPrune (Ye et al. 2025), and SparseVLM (Zhang et al. 2024) prune low-attention tokens based on attention scores, while LLaVA-PruMerge (Shang et al. 2025) clusters similar tokens to retain key visual context with fewer tokens. These methods either rely on attention scores, which are incompatible with FlashAttention (Dao et al. 2022), or involve additional similarity computations, adding substantial overhead. In this work, we propose a simple yet effective pruning method that is fully compatible with FlashAttention and requires negligible extra computation. Moreover, unlike prior works that focus solely on better token selection criteria, we introduce a complementary strategy that explicitly encourages the model to retain essential information in the non-pruned tokens, thereby minimizing the information loss induced by pruning.

3 Method

In this section, we first outline general MLLM architectures (Sec. 3.1). We then introduce progressive question conditioning, which injects question embeddings into ViT layers with gradually increasing frequency to produce question-aware visual features (Sec. 3.2). Next, we introduce an L_2 norm-based pruning strategy that removes background tokens for efficiency. We also propose token focusing, which adapts the model to pruning by encouraging essential information to reside in the retained tokens. (Sec. 3.3). Finally, we combine these components to form TabFlash, an efficient MLLM for table understanding (Sec. 3.4).

3.1 Overall architecture of MLLMs

In this section, we outline the general pipeline of Multimodal Large Language Models (MLLMs). MLLM generates a response given an input image $\mathbf{I}$ and a question $\mathbf{Q}$. An input image $\mathbf{I}$ is first fed into a Vision Transformer (ViT), which extracts a processed set of visual tokens $\mathbf{V}$, which summarizes the image, through multiple attention layers. The process within ViT is formally defined as follows. Provided an input image $\mathbf{I}$, an initial input embedding is generated as $\mathbf{V}_1 = \text{Emb}_v(\mathbf{I}) \in \mathbb{R}^{v \times d}$, where $\text{Emb}_v(\cdot)$ denotes an image embedding layer, v and d denote the number of tokens and ViT feature dimension, respectively. Then, the embedding is passed through multiple ViT layers with index of $l = 1, 2 \ldots, L$, where L denotes the number of ViT layers. In the l-th layer, self-attention is first applied to input $\mathbf{V}_l$, followed by an MLP projection layer, resulting in refined output $\mathbf{V}_{l+1}$, which is fed to the $(l+1)$-th layer as input. The process is defined as:

$$\mathbf{V}_{l+1} \in \mathbb{R}^{v \times d} = \text{MLP}_l\left(\text{Self-Attn}_l\left(\mathbf{V}_l\right)\right). \tag{1}$$

After the L-th layer, $\mathbf{V}_{L+1}$ is generated, which is an image representation extracted by ViT. In the rest of the paper, we denote $\mathbf{V}_{L+1}$ as a visual token set $\mathbf{V}$ for conciseness. Obtained $\mathbf{V}$ are then fed into a projector (*e.g.* MLP (Liu et al. 2023)) which maps visual tokens to the language model (LM) space. Finally, $\mathcal{M}_\theta$, an LM parameterized by θ takes visual tokens $\mathbf{V}$ and a question $\mathbf{Q}$ as input, and generates final response. The model is trained by minimizing the conventional LLM loss, defined as:

$$\mathcal{L}_{\text{llm}} = \text{CE}\left(\mathcal{M}_\theta\left(\hat{\mathbf{y}}|\mathbf{V}, \mathbf{Q}\right)|\mathbf{y}\right), \tag{2}$$

where $\text{CE}(\cdot)$ denotes the cross-entropy loss between the prediction $\hat{\mathbf{y}}$ and the correct response $\mathbf{y}$. Although the architecture has been successful on natural images, we identify two key limitations when applying it to tabular understanding tasks. First, the visual token set $\mathbf{V}$ is generated without considering the question $\mathbf{Q}$. This is especially problematic for table images, where focus on local regions relevant to the question is crucial. Also, as a large number of visual tokens are generated to represent table images in detail, a significant computational burden is induced, as LM generally has quadratic complexity regarding the number of input tokens. To this end, we propose a question-conditioning method to generate $\mathbf{V}$ relevant to the question, and a pruning strategy along with tailored fine-tuning objectives to pursue efficiency while minimizing information loss.

3.2 Progressive Question Conditioning for ViT

Question conditioning for ViT. To obtain a visual token set $\mathbf{V}$ relevant to the question, we additionally condition ViT layers with the question $\mathbf{Q}$, motivated by previous works (Ganz et al. 2024; Abramovich et al. 2024). A question $\mathbf{Q}$ is first converted to embeddings with length of q via an embedding function $\text{Emb}_q(\cdot)$, which is a tokenizer of the LM. Then, a two-layer MLP $\mathcal{P}_l(\cdot)$ projects converted embedding to ViT feature dimension d, resulting in question embedding $\mathbf{Q}_l$ for layer l as follows:

$$\mathbf{Q}_l \in \mathbb{R}^{q \times d} = \mathcal{P}_l\left(\text{Emb}_q\left(\mathbf{Q}\right)\right). \tag{3}$$

Then, generated question embedding $\mathbf{Q}_l$ is concatenated to input embedding $\mathbf{V}_l$ as:

$$\mathbf{V}_l^c \in \mathbb{R}^{(v+q) \times d} = \text{Concat}\left(\left[\mathbf{V}_l, \mathbf{Q}_l\right]\right), \tag{4}$$

forming a combined embedding $\mathbf{V}_l^c$, where v and q denotes number of input and question tokens, respectively. Then, information between tokens is fused through a self-attention operation as:

$$\mathbf{V}_l^{'} \in \mathbb{R}^{(v+q) \times d} = \text{Self-Attn}_l\left(\mathbf{V}_l^c\right), \tag{5}$$

resulting in fused embedding $\mathbf{V}_l^{'}$. Then, only the first v tokens from $\mathbf{V}_l^{'}$ corresponding to original input tokens are selected and fed into the MLP projection layer as:

$$\mathbf{V}_{l+1} \in \mathbb{R}^{v \times d} = \text{MLP}_l\left(\mathbf{V}_l^{'}[0:v]\right), \tag{6}$$

resulting in the layer output $\mathbf{V}_{l+1}$ having the same number of tokens as the layer input $\mathbf{V}_l$. By injecting the question embedding into ViT, the final visual token set $\mathbf{V}$ that is more relevant to the question is obtained.

Progressive question conditioning. Although question conditioning might help obtain a better visual representation, selecting layer l to conduct conditioning is non-trivial. As shown in (Ganz et al. 2024), conditioning on inappropriate layers might even degrade the performance. To this end, we propose *progressive question conditioning*, which injects the question into ViT layers with progressively increasing frequencies. In other words, early ViT layers are intermittently conditioned with a large interval, while the interval decreases as the layer progresses, conditioning late layers more frequently. Such a design choice is grounded on previous observations (Dosovitskiy et al. 2021; Raghu et al. 2021) that early ViT layers are volatile as they focus on details of the image, while latter layers, aggregating global information, are relatively stable. Intuitively, progressive conditioning allows stable injection of question information by adjusting the conditioning frequency proportional to each layer's capacity to handle the question. Note that question conditioning adds only a negligible total computation cost of 0.4%, making it highly efficient.

3.3 Token Focusing with Background Pruning

Background token pruning. Conventional MLLMs (Chen et al. 2024b; Zhao et al. 2024) produce up to 2–3k visual tokens, incurring high computational cost due to the quadratic complexity of LMs. However, tabular images are highly redundant, with a large portion of tokens representing background. We observe that the L_2 norm of visual tokens effectively distinguishes content from background. In detail, high norms align with content regions, while low norms indicate background (Fig. 3, left). Based on this observation, we propose a background pruning strategy. Given pruning rate p, we take $N_r = \lfloor (1-p) \cdot v \rfloor$ tokens with the highest norms to get retained set $\mathbf{V}_r$ as:

$$\mathbf{V}_r = \{\mathbf{v}_i \mid i \in \text{Top-}k\left(\|\mathbf{V}\|_2; N_r\right)\}, \tag{7}$$

where $\|\mathbf{V}\|_2 = \{\|\mathbf{v}_i\|_2\}_{i=1}^{v}$ is a set consisting of L_2 norms of tokens $\mathbf{v}_i \in \mathbf{V}$, and Top-$k(\cdot; N_r)$ is an operation returning

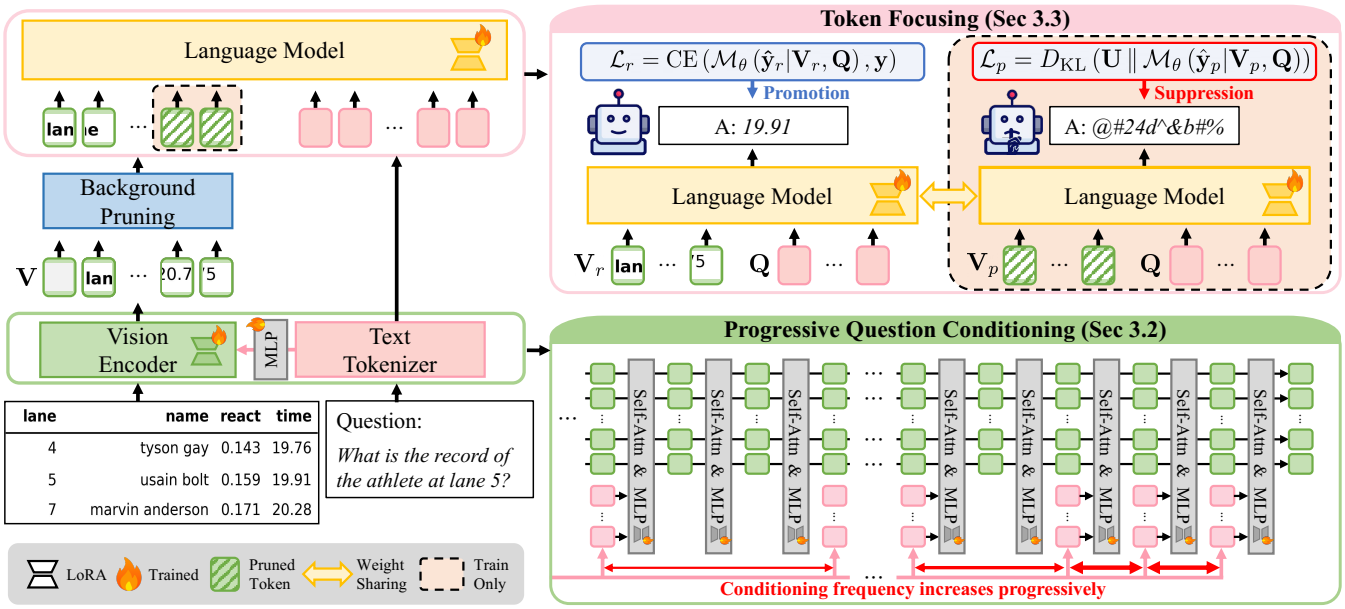

Figure 2: **Overall pipeline of TabFlash.** Progressive question conditioning injects question information into ViT layers with a progressively increasing frequency, producing a question-relevant visual token set $\mathbf{V}$ (Sec. 3.2). The tokens are divided into a pruned set $\mathbf{V}_p$ and a retained set $\mathbf{V}_r$, where only $\mathbf{V}_r$ is used during inference for efficiency. To concentrate information in $\mathbf{V}_r$, token focusing encourages accurate prediction with $\mathbf{V}_r$ while suppressing prediction using $\mathbf{V}_p$ (Sec. 3.3).

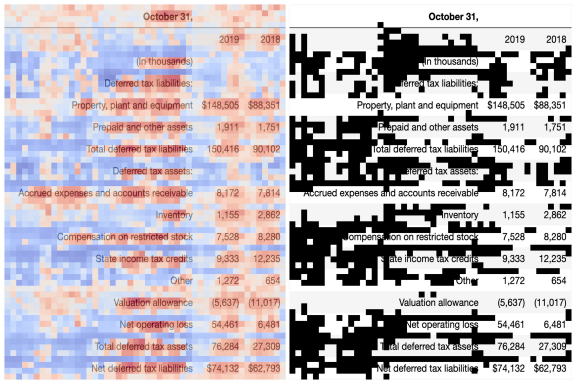

Figure 3: **Visualization of L_2 norms of ViT output tokens (left) and norm-based pruning results (right).** Red and blue color denotes high and low L_2 norms, respectively. 30% of tokens with the lowest norms are pruned ($p = 0.3$).

indices of N_r elements with highest values. The remaining tokens form the pruned token set $\mathbf{V}_p$, formally defined as:

$$\mathbf{V}_p = \mathbf{V} \setminus \mathbf{V}_r. \tag{8}$$

During inference, only the retained set $\mathbf{V}_r$ is fed to LM. Visualization of the retained set $\mathbf{V}_r$ (Fig. 3, right) shows successful removal of background tokens. Unlike attention-based or similarity-based pruning, our method is compatible with FlashAttention (Dao et al. 2022) and avoids costly similarity map construction, making it highly efficient.

Token focusing. While background pruning successfully discards background tokens, simply discarding the pruned set $\mathbf{V}_p$ and using the retained set $\mathbf{V}_r$ for inference leads to a significant performance drop (Tab. 4, row 4). A closer look reveals that the model can still answer questions to some extent using only $\mathbf{V}_p$ (Tab. 5), indicating that important information is still being stored in the pruned tokens. This behavior is undesirable since only the retained tokens $\mathbf{V}_r$ are used during inference, meaning any useful information in $\mathbf{V}_p$ is lost, aggravating the performance degradation. To address this, we introduce *token focusing*, a novel training strategy that encourages the model to focus important information in the retained tokens $\mathbf{V}_r$. Token focusing explicitly promotes information retention on $\mathbf{V}_r$, while discouraging it on $\mathbf{V}_p$. Specifically, we define the token promotion loss $\mathcal{L}_r$ to encourage accurate predictions based solely on $\mathbf{V}_r$:

$$\mathcal{L}_r = \mathrm{CE}\left(\mathcal{M}_\theta\left(\hat{\mathbf{y}}_r|\mathbf{V}_r, \mathbf{Q}\right), \mathbf{y}\right), \tag{9}$$

where CE denotes the cross-entropy loss, $\hat{\mathbf{y}}_r$ is prediction with $\mathbf{V}_r$ and $\mathbf{y}$ is the target answer. On the other hand, to suppress the information retention in $\mathbf{V}_p$, we define a token suppression loss $\mathcal{L}_p$ as the KL divergence between the model's prediction based on $\mathbf{V}_p$ and a uniform distribution $\mathbf{U}$ over the vocabulary:

$$\mathcal{L}_p = D_{\mathrm{KL}}\left(\mathbf{U} \| \mathcal{M}_\theta\left(\hat{\mathbf{y}}_p|\mathbf{V}_p, \mathbf{Q}\right)\right), \tag{10}$$

where $\hat{\mathbf{y}}_p$ is prediction with $\mathbf{V}_p$. Minimizing $\mathcal{L}_p$ penalizes meaningful prediction from $\mathbf{V}_p$, thereby discouraging the storage of meaningful information in pruned tokens. Combining both objectives, the token focusing loss is defined as:

$$\mathcal{L}_{\mathrm{tf}} = \mathcal{L}_r + \lambda \cdot \mathcal{L}_p, \tag{11}$$

Method	LM Size	In-domain					Out-domain		Avg.
		TABMWP	WTQ	HiTab	TAT-QA	FeTaQA	AIT-QA	TabMCQ	
Proprietary MLLM									
GPT-4o	-	47.5	25.7	10.2	25.1	10.8	18.2	40.0	25.4
Gemini 2.5 Pro (Comanici et al. 2025)	-	48.1	49.2	24.2	19.6	5.4	37.6	18.2	28.9
GPT-4V†	-	60.0	48.0	27.5	32.5	11.0	62.5	66.0	43.9
Open-Source MLLM									
LLaVA-1.5 (Liu et al. 2024)	7B	6.1	1.2	2.0	3.0	8.2	-	-	-
Vary-toy (Wei et al. 2024)	1.8B	4.4	8.0	3.4	8.8	2.4	9.4	-	-
Monkey (Li et al. 2024)	7B	13.3	19.1	6.4	12.3	3.4	-	18.9	-
TabPedia (Zhao et al. 2024)	7B	12.3	20.4	1.2	9.7	12.5	17.2	1.0	10.6
mPlug-DocOwl1.5 (Hu et al. 2024)	7B	11.4	26.8	11.1	12.4	3.6	46.2	3.2	16.4
Table-LLaVA (Zheng et al. 2024)	7B	57.8	18.4	10.1	12.8	25.6	5.5	44.5	25.0
Table-LLaVA (Zheng et al. 2024)	13B	59.8	20.4	10.9	15.7	28.0	6.1	51.5	27.5
SynTab-LLaVA (Zhou et al. 2025)	7B	88.3	39.6	35.7	51.9	35.5	28.6	70.6	50.0
InternVL-2.5 (Chen et al. 2024b)	3B	93.1	45.2	53.7	55.7	31.8	**58.7**	64.1	57.5
TabFlash	3B	**93.7**	**46.4**	**60.5**	**59.9**	**36.1**	54.9	**71.9**	**60.5**
Low-cost models‡ (<5 TFLOPs)									
InternVL-2.5 (Chen et al. 2024b)	1B	86.3	30.5	32.3	35.5	26.6	39.8	50.0	43.0
TabFlash	1B	88.5	32.1	40.9	44.8	32.9	41.9	57.8	48.4

Table 1: **Results on table question-answering benchmarks.** Best results among open-source MLLMs highlighted **bold**. Avg.: performance averaged over seven benchmarks. Out-domain: model is not trained with the training set of the dataset. †: evaluation results on subset (Zheng et al. 2024). ‡: models with exceptionally low cost ($<$ 5 TFLOPs). See Tab. 2 for cost analysis.

where λ is a hyperparameter. Unlike existing pruning methods that aim to find better $\mathbf{V}_p$ with minimal information loss, token focusing takes a complementary approach: it regularizes the location of information storage itself, guiding the model to preserve critical content within $\mathbf{V}_r$ rather than $\mathbf{V}_p$. Our analysis (Sec. 5.2) demonstrates that token focusing contributes to the transfer of information from $\mathbf{V}_p$ to $\mathbf{V}_r$, thereby minimizing the information loss by pruning.

3.4 TabFlash

TabFlash. Combining progressive question conditioning and pruning with token focusing, we present TabFlash. Equipped with both methods, a compact and informative token set $\mathbf{V}_r$ is obtained, improving both the effectiveness and efficiency. Fig. 2 outlines the overall pipeline of TabFlash.

Training process. We train TabFlash in two stages. First, the model is trained for 3 epochs using progressive question conditioning with conventional LLM loss $\mathcal{L}_{\text{llm}}$ (Eq. (2)). Then, it is trained for an additional epoch with both progressive question conditioning and background pruning applied, utilizing the token focusing loss $\mathcal{L}_{\text{tf}}$ (Eq. (11)). The 3B variant model is efficiently trained within 22 hours using 8 NVIDIA H200 GPUs.

4 Experiments

4.1 Implementation Details

We develop TabFlash by fine-tuning the 1B and 3B variants of InternVL-2.5 (Chen et al. 2024b). Following Table-LLaVA (Zheng et al. 2024), we train on the MMTab-pre and MMTab-instruct datasets, totaling 383k samples. LoRA (Hu et al. 2022) with rank 16 is applied to both the ViT and LLM, while all other parameters remain frozen. Training is conducted for 4 epochs with a learning rate of $4e{-}5$, $\lambda = 2e{-}4$, and pruning rate $p = 0.3$. The exact configuration of question conditioning layers is in the supplementary material.

4.2 Datasets and Evaluation Metrics

We evaluate the performance on seven table question-answering datasets, TABMWP (Lu et al. 2023), WTQ (Pasupat and Liang 2015), HiTab (Cheng et al. 2021), TAT-QA (Zhu et al. 2021), FeTaQA (Nan et al. 2022), AIT-QA (Katsis et al. 2021), and TabMCQ (Jauhar, Turney, and Hovy 2016) over 17.9k samples in total. For evaluation protocol and metrics, we directly follow the settings of Table-LLaVA (Zheng et al. 2024) without any modification. Accuracy is adopted as a metric for all datasets except for FeTaQA, adopting BLEU (Papineni et al. 2002) as a metric.

4.3 Main Results

In Tab. 1, we compare TabFlash with proprietary and open-source MLLMs on seven table QA benchmarks. While QA is our main focus, we also report results on fact verification and table-to-text generation in the supplementary material.

Results TabFlash (3B) achieves the highest average performance (60.5), outperforming all open-source models on six of seven benchmarks and ranking second on AIT-QA. The 1B variant also performs well, surpassing most prior models with an average score of 48.4. Importantly, this is achieved with significantly lower computational cost, as detailed in the next section. Notably, TabFlash also outperforms proprietary models (*e.g.* GPT-4V, Gemini 2.5 Pro) on average, highlighting both the difficulty of table understanding and the effectiveness of our approach.

Method	LM Size	TFLOPs ↓	Memory↓	Avg.↑
TabPedia	7B	27.08	22.7G	10.6
DocOwl1.5	7B	26.60	24.0G	16.4
Table-LLaVA	7B	10.42	15.0G	25.0
Table-LLaVA	13B	18.47	28.1G	27.5
SynTab-LLaVA	7B	15.21	16.4G	50.0
InternVL-2.5	3B	14.23	24.7G	57.5
TabFlash	3B	10.38	17.3G	**60.5**
Low-cost models (<5 TFLOPs)				
InternVL-2.5	1B	2.59	18.5G	43.0
TabFlash	1B	**1.78**	**11.2G**	48.4

Table 2: **Cost Analysis.** Memory: Peak GPU memory usage.

Injection layers	Interval	# cond. layers	Avg.
All (1-24)	1	24	43.4
Early (1-8)	1	8	44.8
Mid (9-16)	1	8	48.0
Late (17-24)	1	8	48.8
Sparse	2	12	48.0
	3	8	48.6
	4	6	49.0
Progressive (Ours)	Progressive	6	**50.3**

Table 3: **Ablation on progressive question conditioning.** # cond. layers: total # layers conditioning is applied to.

4.4 Cost Analysis

In Tab. 2, we report the LLM TFLOPs and peak GPU memory usage of TabFlash and previous open-source MLLMs. TabFlash (3B) achieves the best overall performance while maintaining high efficiency, requiring only 10.38 TFLOPs and 17.3 GB of memory. Compared to the second-best model, InternVL-2.5 (3B), TabFlash reduces FLOPs by 27% and memory usage by 30%, while achieving a 3-point gain in accuracy. The 1B variant of TabFlash further improves computational efficiency, consuming just 1.78 TFLOPs and 11.2 GB of memory, and still outperforms five of the seven baselines. Compared to SynTab-LLaVA with a difference of 1.6%p in accuracy, TabFlash (1B) requires 88% fewer FLOPs and 32% less memory, highlighting its exceptional cost-effectiveness for resource-constrained scenarios. Latency analysis is provided in the supplementary material.

5 Analysis

In this section, we present various analyses to investigate the effect of each component. All experiments report the average accuracy achieved by TabFlash (1B). Detailed results for each dataset are provided in the supplementary material.

5.1 Ablation Study

Progressive question conditioning. In Tab. 3, we compare conditioning strategies based on injection layer choices. We evaluate conditioning at all layers ('All'), as well as selectively at early ('Early'), mid ('Mid'), and late ('Late') layers, all with a fixed interval of 1. We then consider sparsely conditioning at every 2, 3, or 4 layers ('Sparse'). Finally, we propose conditioning with a progressively decreasing interval, where the conditioning frequency increases as the layer progresses ('Progressive'). Please refer to the supplementary material for more detailed configurations. Results suggest that early layers are particularly sensitive to conditioning, and overly frequent injection degrades performance, making stable information integration difficult. Nevertheless, our method achieves the best performance, demonstrating its robustness in injecting question information across layers.

Method	TFLOPs ↓	Pruning	Avg.↑
Upper bound (unpruned)	2.59		50.3 (100%)
Ours (both $\mathcal{L}_p$, $\mathcal{L}_r$)	1.78	✓	48.4 (96%)
- without $\mathcal{L}_p$	1.78	✓	47.3 (94%)
- without $\mathcal{L}_r$	1.78	✓	37.2 (74%)

Table 4: **Ablation on token focusing loss.** ✓: pruning used at inference. Percentages are relative to the unpruned model.

Training Loss	Token set for inference	Avg.	Δ
$\mathcal{L}_r$	$\mathbf{V}_r$	47.3	33.8
	$\mathbf{V}_p$	13.5	
$\mathcal{L}_r, \mathcal{L}_p$	$\mathbf{V}_r$	48.4	48.4
	$\mathbf{V}_p$	0.0	

Table 5: **Performance comparison using $\mathbf{V}_r$ and $\mathbf{V}_p$ for inference.** Δ denotes the difference in average performance between the retained set $\mathbf{V}_r$ and pruned set $\mathbf{V}_p$ for inference.

Token focusing. In Tab. 4, we present ablation results for token focusing. Compared to TabFlash without pruning, applying pruning with the proposed token focusing reduces FLOPs by 31%, while limiting the performance degradation to only 3.8%. Removing the token suppression loss ($\mathcal{L}_p$) leads to an additional performance drop of 1.1 points, showing its importance. Furthermore, removing the token promotion loss ($\mathcal{L}_r$) results in a substantial decline in performance. These results highlight the significance of token focusing, demonstrating the necessity of a proper fine-tuning strategy facilitating the model's adaptation to pruning results by enforcing information concentration on the retained token set.

5.2 Effect of Token Focusing

To investigate whether the suppression loss operates as desired, we compare the performance of models trained with and without the suppression loss $\mathcal{L}_p$, evaluating when different token sets are provided to the LM for inference (Tab. 5). When trained solely with the token promotion loss $\mathcal{L}_r$, accuracy of 13.5 is achieved even using the pruned set $\mathbf{V}_p$, indicating the presence of useful information in $\mathbf{V}_p$. Adding the suppression loss $\mathcal{L}_p$ addresses this by penalizing the model's ability to answer correctly from $\mathbf{V}_p$, thereby concentrating valuable information in the retained set $\mathbf{V}_r$. Thus, the model trained with $\mathcal{L}_p$ shows 1.1 points higher accuracy with $\mathbf{V}_r$

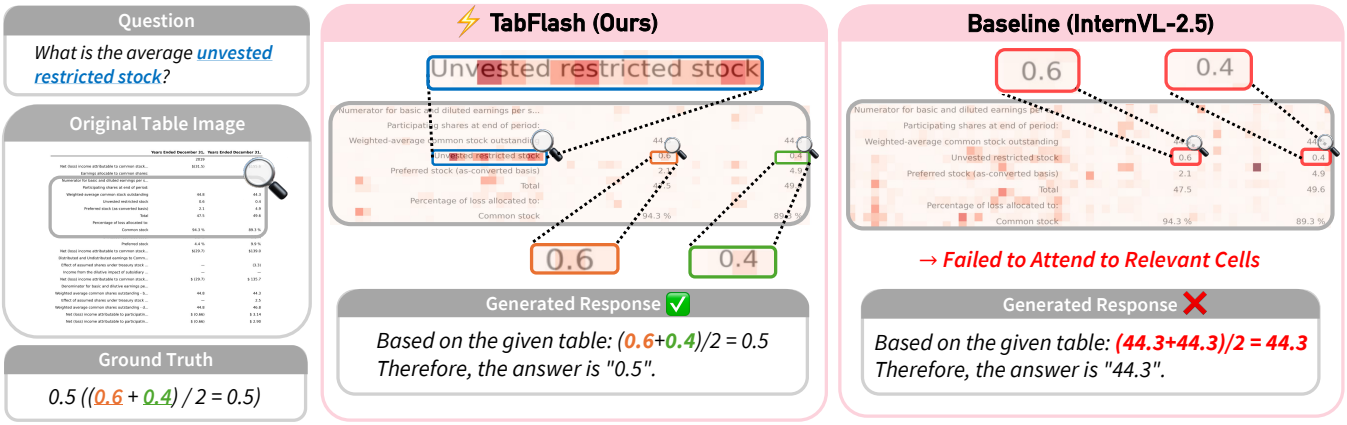

Figure 4: **Qualitative results.** Near-white regions indicate low attention, while stronger red colors represent higher attention scores. Best viewed when zoomed in. Please refer to the supplementary material for further qualitative results.

Pruning rate p	TFLOPs ↓	Memory↓	Avg.↑
0.0	2.59 (100%)	18.7G (100%)	50.3 (100%)
0.1	2.31 (89%)	15.3G (82%)	49.4 (98%)
0.2	2.03 (78%)	12.4G (66%)	48.5 (96%)
0.3	1.78 (69%)	11.2G (60%)	48.4 (96%)
0.4	1.52 (59%)	11.1G (59%)	46.7 (93%)
0.5	1.28 (49%)	11.0G (59%)	44.0 (87%)

Table 6: **Analysis on pruning rate p.** Percentages are relative to the unpruned baseline (first row).

than without $\mathcal{L}_p$. Also, the gap between $\mathbf{V}_r$ and $\mathbf{V}_p$ widens from 33.8 to 48.4 points, indicating information transfer from $\mathbf{V}_p$ to $\mathbf{V}_r$. These results validate the effectiveness of token focusing for retaining information on $\mathbf{V}_r$, making the model more compatible with the pruning strategy.

5.3 Analysis on Pruning Rate p

Tab. 6 reports TFLOPs, peak memory, and accuracy across pruning rates (p). Without pruning ($p = 0.0$), the model achieves its full performance of 50.3. At $p = 0.1$, the model retains 98.2% of its original performance, while reducing TFLOPs and memory usage by 11% and 18%, respectively. A more aggressive pruning rate of $p = 0.5$ yields substantial efficiency gains, cutting TFLOPs by 51% and memory by 41%. These results show that the pruning rate offers a tunable trade-off between performance and efficiency. We set the default rate to $p = 0.3$, achieving 31% TFLOPs and 40% memory reduction with only a modest performance drop.

5.4 Performance Comparison by Table Size

In Tab. 7, we compare the accuracy of TabFlash with the baseline architecture, InternVL, across three subsets divided by image sizes: small, medium, and large. Difficulty increases proportionally to the image size, as the portion of the question-relevant region gets smaller in larger images. The performance gap between baseline and TabFlash widens as the size of the image enlarges, reaching the 15% gap in the 'large' group. These results highlight TabFlash's capability in capturing fine-grained details in larger images with informative visual tokens.

Method	Small	Medium	Large
InternVL-2.5 (1B)	85.1	64.1	29.9
TabFlash (1B)	87.6 (+3%)	67.6 (+5%)	34.4 (+15%)

Table 7: **Performance by table size.** Small, Medium, and Large each include equal-sized subsets sorted by image size.

5.5 Qualitative Analysis

Fig. 4 shows the average attention each visual token receives during generation, where higher scores indicate stronger information extraction. TabFlash concentrates attention on the question-relevant rows, which appear in only a small portion of the image, whereas the baseline spreads attention broadly and overlooks key regions. This focus enables TabFlash to identify the correct evidence row ("Unvested restricted stock") and use the required values ('0.6', '0.4') for the final calculation. Meanwhile, the baseline attends to irrelevant rows, leading to an incorrect result (*e.g.* '44.3'). These findings show that progressive question conditioning produces question-aware visual features and more accurate attention.

6 Conclusion

We present TabFlash, an efficient MLLM for table understanding. Progressive question conditioning injects the question into ViT layers at progressively increasing frequency. Background pruning discards redundant tokens based on the norm of each token, effectively removing uninformative regions. Token focusing minimizes the information loss induced by pruning by enforcing the concentration of essential information on retained tokens. Together, TabFlash achieves the state-of-the-art performance with significantly reduced computational cost.

Acknowledgments

This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. RS-2024-00443251, Accurate and Safe Multimodal, Multilingual Personalized AI Tutors, 40%), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. RS-2024-00457882, AI Research Hub Project, 30%), and National Supercomputing Center with supercomputing resources including technical support (KSC-2025-CRE-0085, 30%).

References

Abramovich, O.; Nayman, N.; Fogel, S.; Lavi, I.; Litman, R.; Tsiper, S.; Tichauer, R.; Appalaraju, S.; Mazor, S.; and Manmatha, R. 2024. VisFocus: Prompt-guided vision encoders for OCR-free dense document understanding. In *ECCV*.

Akhtar, M.; Cocarascu, O.; and Simperl, E. 2022. PubHealthTab: A public health table-based dataset for evidence-based fact checking. In *NAACL Findings*.

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *NeurIPS*.

Bolya, D.; Fu, C.-Y.; Dai, X.; Zhang, P.; Feichtenhofer, C.; and Hoffman, J. 2023. Token merging: Your vit but faster. *ICLR*.

Chen, L.; Zhao, H.; Liu, T.; Bai, S.; Lin, J.; Zhou, C.; and Chang, B. 2024a. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *ECCV*.

Chen, W.; Wang, H.; Chen, J.; Zhang, Y.; Wang, H.; Li, S.; Zhou, X.; and Wang, W. Y. 2020. Tabfact: A large-scale dataset for table-based fact verification. *ICLR*.

Chen, Z.; Wang, W.; Cao, Y.; Liu, Y.; Gao, Z.; Cui, E.; Zhu, J.; Ye, S.; Tian, H.; Liu, Z.; et al. 2024b. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Cheng, Z.; Dong, H.; Wang, Z.; Jia, R.; Guo, J.; Gao, Y.; Han, S.; Lou, J.-G.; and Zhang, D. 2021. Hitab: A hierarchical table dataset for question answering and natural language generation. *ACL*.

Choi, J.; Lee, S.; Chu, J.; Choi, M.; and Kim, H. J. 2024. vid-tldr: Training free token merging for light-weight video transformer. In *CVPR*.

Comanici, G.; Bieber, E.; Schaekermann, M.; Pasupat, I.; Sachdeva, N.; Dhillon, I.; Blistein, M.; Ram, O.; Zhang, D.; Rosen, E.; et al. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; and Ré, C. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *NeurIPS*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.

Ganz, R.; Kittenplon, Y.; Aberdam, A.; Ben Avraham, E.; Nuriel, O.; Mazor, S.; and Litman, R. 2024. Question aware vision transformer for multimodal reasoning. In *CVPR*.

Gupta, V.; Mehta, M.; Nokhiz, P.; and Srikumar, V. 2020. INFOTABS: Inference on tables as semi-structured data. *ACL*.

Hu, A.; Xu, H.; Ye, J.; Yan, M.; Zhang, L.; Zhang, B.; Li, C.; Zhang, J.; Jin, Q.; Huang, F.; et al. 2024. mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. *EMNLP*.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. In *ICLR*.

Jauhar, S. K.; Turney, P.; and Hovy, E. 2016. Tabmcq: A dataset of general knowledge tables and multiple-choice questions. *arXiv preprint arXiv:1602.03960*.

Katsis, Y.; Chemmengath, S.; Kumar, V.; Bharadwaj, S.; Canim, M.; Glass, M.; Gliozzo, A.; Pan, F.; Sen, J.; Sankaranarayanan, K.; et al. 2021. Ait-qa: Question answering dataset over complex tables in the airline industry. *NAACL*.

Lebret, R.; Grangier, D.; and Auli, M. 2016. Neural text generation from structured data with application to the biography domain. *EMNLP*.

Lee, S.; Choi, J.; and Kim, H. J. 2024. Multi-criteria token fusion with one-step-ahead attention for efficient vision transformers. In *CVPR*.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

Li, Z.; Yang, B.; Liu, Q.; Ma, Z.; Zhang, S.; Yang, J.; Sun, Y.; Liu, Y.; and Bai, X. 2024. Monkey: Image resolution and text label are important things for large multi-modal models. In *CVPR*.

Liang, Y.; Ge, C.; Tong, Z.; Song, Y.; Wang, J.; and Xie, P. 2022. Not all patches are what you need: Expediting vision transformers via token reorganizations. *ICLR*.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024. Improved baselines with visual instruction tuning. In *CVPR*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. *NeurIPS*.

Lu, P.; Qiu, L.; Chang, K.-W.; Wu, Y. N.; Zhu, S.-C.; Rajpurohit, T.; Clark, P.; and Kalyan, A. 2023. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *ICLR*.

Nan, L.; Hsieh, C.; Mao, Z.; Lin, X. V.; Verma, N.; Zhang, R.; Kryściński, W.; Schoelkopf, H.; Kong, R.; Tang, X.; et al. 2022. FeTaQA: Free-form table question answering. *TACL*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Park, J.; Bae, M.; Ko, D.; and Kim, H. J. 2024. Llamo: Large language model-based molecular graph assistant. In *NeurIPS*.

Park, J.; Na, J.; Kim, J.; and Kim, H. J. 2025. DeepVideo-R1: Video Reinforcement Fine-Tuning via Difficulty-aware Regressive GRPO. In *NeurIPS*.

Pasupat, P.; and Liang, P. 2015. Compositional semantic parsing on semi-structured tables. *ACL*.

Raghu, M.; Unterthiner, T.; Kornblith, S.; Zhang, C.; and Dosovitskiy, A. 2021. Do vision transformers see like convolutional neural networks? In *NeurIPS*.

Shang, Y.; Cai, M.; Xu, B.; Lee, Y. J.; and Yan, Y. 2025. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. In *ICCV*.

Wei, H.; Kong, L.; Chen, J.; Zhao, L.; Ge, Z.; Yang, J.; Sun, J.; Han, C.; and Zhang, X. 2024. Vary: Scaling up the vision vocabulary for large vision-language model. In *ECCV*.

Wiseman, S.; Shieber, S. M.; and Rush, A. M. 2017. Challenges in data-to-document generation. *EMNLP*.

Ye, W.; Wu, Q.; Lin, W.; and Zhou, Y. 2025. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *AAAI*.

Zhang, Y.; Fan, C.-K.; Ma, J.; Zheng, W.; Huang, T.; Cheng, K.; Gudovskiy, D.; Okuno, T.; Nakata, Y.; Keutzer, K.; et al. 2024. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.

Zhao, W.; Feng, H.; Liu, Q.; Tang, J.; Wu, B.; Liao, L.; Wei, S.; Ye, Y.; Liu, H.; Zhou, W.; et al. 2024. Tabpedia: Towards comprehensive visual table understanding with concept synergy. *NeurIPS*.

Zheng, M.; Feng, X.; Si, Q.; She, Q.; Lin, Z.; Jiang, W.; and Wang, W. 2024. Multimodal Table Understanding. In *ACL*.

Zhou, B.; Gao, Z.; Wang, Z.; Zhang, B.; Wang, Y.; Chen, Z.; and Xie, H. 2025. SynTab-LLaVA: Enhancing Multimodal Table Understanding with Decoupled Synthesis. In *CVPR*.

Zhu, F.; Lei, W.; Huang, Y.; Wang, C.; Zhang, S.; Lv, J.; Feng, F.; and Chua, T.-S. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. *ACL*.