HiRED: Attention-Guided Token Dropping for Efficient Inference of High-Resolution Vision-Language Models

Kazi Hasan Ibn Arif[1], JinYi Yoon[1], Dimitrios S. Nikolopoulos[1], Hans Vandierendonck[2], Deepu John[3], Bo Ji[1]

[1]Virginia Tech, Blacksburg, VA, USA
[2]Queen's University Belfast, Belfast, UK
[3]University College Dublin, Dublin, Ireland
{hasanarif, jinyiyoon, dsn}@vt.edu, h.vandierendonck@qub.ac.uk, deepu.john@ucd.ie, boji@vt.edu

Abstract

High-resolution Vision-Language Models (VLMs) are widely used in multimodal tasks to enhance accuracy by preserving detailed image information. However, these models often generate an excessive number of visual tokens due to the need to encode multiple partitions of a high-resolution image input. Processing such a large number of visual tokens poses significant computational challenges, particularly for resource-constrained commodity GPUs. To address this challenge, we propose *High-Resolution Early Dropping (HiRED)*, a plug-and-play token-dropping method designed to operate within a fixed token budget. HiRED leverages the attention of `CLS` token in the vision transformer (ViT) to assess the visual content of the image partitions and allocate an optimal token budget for each partition accordingly. The most informative visual tokens from each partition within the allocated budget are then selected and passed to the subsequent Large Language Model (LLM). We showed that HiRED achieves superior accuracy and performance, compared to existing token-dropping methods. Empirically, HiRED-20% (i.e., a 20% token budget) on LLaVA-Next-7B achieves a 4.7× increase in token generation throughput, reduces response latency by 78%, and saves 14% of GPU memory for single inference on an NVIDIA TESLA P40 (24 GB). For larger batch sizes (e.g., 4), HiRED-20% prevents out-of-memory errors by cutting memory usage by 30%, while preserving throughput and latency benefits.

Code — https://github.com/hasanar1f/HiRED

1 Introduction

Vision-Language Models (VLMs), such as GPT-4v (Achiam et al. 2024), Gemini Pro (Reid et al. 2024), LLaVA (Liu et al. 2023), and Qwen-VL (Bai et al. 2023), have emerged as remarkable multimodal models that learn from visual and textual data. However, these VLMs inherently work for low-resolution images only and would lose fine-grained visual information if applied to high-resolution images (Zhang et al. 2024a; Dong et al. 2024). To address this issue, recent VLMs, referred to as high-resolution VLMs, employ dynamic partitioning to encode high-resolution images (Liu et al. 2024a; Dong et al. 2024; Li et al. 2024; Liu et al. 2024c; Lin et al. 2023; Chen, Pekis, and Brown 2024).

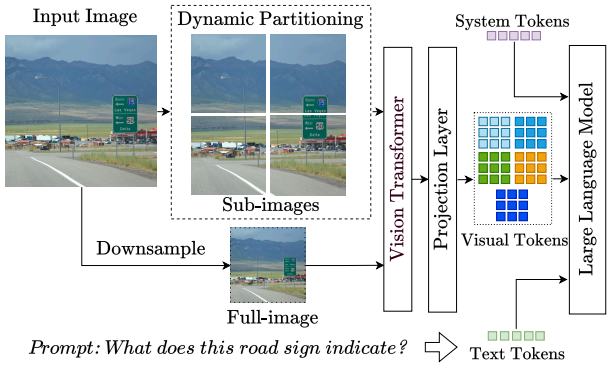


Figure 1: Inference steps of LLaVA-Next (Liu et al. 2024a) for a high-resolution VLM with dynamic partitioning.

A typical inference pipeline of high-resolution VLMs with dynamic partitioning is illustrated in Fig. 1. Specifically, a high-resolution input image is partitioned into multiple sub-images (e.g., four sub-images for a square image in LLaVA-Next); a downsampled version of the original image, referred to as the full-image, is also included. Subsequently, a vision encoder such as Vision Transformers (ViTs) encodes each low-resolution image partition into image features, which are then converted to visual tokens in the text embedding space through a lightweight Projection Layer. These visual tokens are concatenated and fed into a Large Language Model (LLM) (along with text tokens and system tokens) to generate the final response. Here, the full-image and the sub-images (commonly referred to as image partitions in this paper) have different amounts of visual content and thus, exhibit different degrees of importance. While the full-image captures the global context of the original image, each sub-image is for a more detailed local representation of corresponding specific areas. This multi-partitioning approach enables the inclusion of more visual details, which can significantly boost accuracy. For example, accuracy can be improved by 15% when the image resolution is increased from 336×336 to 1344×1344 (McKinzie et al. 2025).

However, due to the need to encode multiple image partitions, high-resolution VLMs often generate 3-10× more visual tokens than their low-resolution counterparts (Dong et al. 2024; Hu et al. 2024). Such excessive visual tokens

Method	High Resolution	Token Budget	Early Dropping	Task Coverage
FastV (Chen et al. 2025b)	✗	✓	✗	✓
FlexAttention (Li et al. 2025)	✓	✗	✗	✓
TokenCorrCompressor (Zhang et al. 2024c)	✓	✗	✓	✗
PruMerge (Shang et al. 2024)	✗	✗	✓	✓
HiRED (Ours)	✓	✓	✓	✓

Table 1: Comparison between our HiRED and existing methods.

result in lower inference throughput, increased generation latency, and higher GPU memory usage. Furthermore, depending on downstream tasks, the number of visual tokens required to represent an image also varies significantly (Cai et al. 2024). However, most commodity GPUs, such as the Jetson Orin NX (8 or 16 GB) and NVIDIA Tesla T4 (16 GB), have limited computational cores and memory. The quadratic complexity of transformers (Vaswani 2017) makes it challenging to process a large number of tokens on these GPUs. In addition, increased key-value (KV) cache size due to storing token embeddings at runtime could cause out-of-memory issues. Therefore, controlling and optimizing the number of visual tokens is essential to meet the system resource constraints. Although traditional optimization techniques (e.g., model quantization, weight pruning, and lightweight architectures) can reduce model size, they do not address the critical issue of excessive visual tokens.

We aim to achieve efficient inference of high-resolution VLMs through strategic dropping of excessive visual tokens. Such token-dropping schemes are expected to offer four desired properties: (i) *Supporting high-resolution*: plug-and-play integration (i.e., without model training and architectural changes) that promotes easy adoption with existing high-resolution VLMs while maintaining superior accuracy; (ii) *Controlling token budget*: having control over the number of visual tokens fed into the LLM to enable efficient inference under various resource constraints and task requirements; (iii) *Facilitating early dropping*: dropping tokens in the image encoding stage (i.e., before the generation phase using LLM) to reduce input length and enhance computational efficiency; and (iv) *Wide task coverage*: covering a wide range of vision-language tasks (vision question answering, image captioning, document understanding, etc.).

The recent few months have witnessed exciting progress towards the above goals (Chen et al. 2025b; Li et al. 2025; Zhang et al. 2024c; Shang et al. 2024). However, none of these works achieve all the aforementioned properties, which are highly desired for efficient high-resolution VLM inference (see Table 1 for a summary and Section 2 for a detailed discussion).

Contributions. Our work bridges this critical gap and makes the following main contributions:

We propose *High-Resolution Early Dropping (HiRED)*, a plug-and-play token-dropping framework for efficient inference of high-resolution VLMs. HiRED enables attention-guided early dropping of visual tokens under resource constraints and covers a wide range of multimodal tasks. *To the best of our knowledge, HiRED is the first framework that achieves all of these desired properties.*

To realize HiRED, our key design leverages two crucial insights from attention patterns in ViT. *First*, class token (`CLS`) to patch token attentions from initial ViT layers are closely correlated with the visual contents and can be used to identify the main objects and irrelevant backgrounds in an image. To allocate a larger budget to a partition with more content, we introduce the *visual content score* (which represents the amount of visual content a partition carries) as the token budget for each sub-image. Second, we observe that `CLS`-attention (attention scores between the class token and patch tokens of ViT) from the final layers indicate the informativeness of patch tokens. Therefore, we use the `CLS`-attention (aggregated across multiple heads) from the final layer as *feature importance score* and select tokens with the highest feature importance score within the allocated budget. *By leveraging such `CLS`-attention patterns in ViT, we design a lightweight yet efficient algorithm for budget allocation and token dropping, two key components of HiRED.*

Finally, we implement HiRED on three popular open-source VLMs: LLaVA-Next (Liu et al. 2024a), LLaVA (Liu et al. 2023), and ShareGPT4V (Chen et al. 2025a) and evaluate the accuracy for eight tasks. Our experimental results show that HiRED-20% (i.e., the budget is set to 20% of the total number of tokens) on LLaVA-Next-7B (a high-resolution VLM) achieves a 4.7× increase in token generation throughput (2.30 vs. 0.49 tokens/sec), reduces response latency by 78% (4.21 vs. 19.49 seconds), and saves 14% of GPU memory (13.76 vs. 16.04 GB) for single inference on an NVIDIA TESLA P40 GPU. For larger batch sizes (e.g., 4), where the 24 GB GPU encounters out-of-memory (OOM) errors with the full token budget, HiRED-20% reduces memory usage by 30% (16.99 GB) while maintaining the throughput and latency improvements. Moreover, HiRED achieves significantly higher accuracy than previous early-dropping methods, such as PruMerge and PruMerge+ (Shang et al. 2024) across various tasks.

2 Related Work

We categorize highly related works into three groups.

Lightweight Architectures. Traditional methods often aim to downsize VLMs by reducing the model size, such as LLaVA-Phi-2.7B (Zhu et al. 2024), TinyLLaVA-3.1B (Zhou et al. 2024), and MobileVLM-3B (Chu et al. 2023). However, these approaches significantly compromise reasoning capabilities due to the substantial reduction in model param-

Budget	*Full*	*TL*	*TR*	*BL*	*BR*
10% (= 288 tokens)	37	12	29	84	126
20% (= 576 tokens)	94	40	61	148	234

Table 2: Distribution of the top (10% and 20%) visual tokens for the image partitions shown in Fig. 1. Here, *TL*, *TR*, *BL*, and *BR* represent the top-left, top-right, bottom-left, and bottom-right corners of the image partitions, respectively.

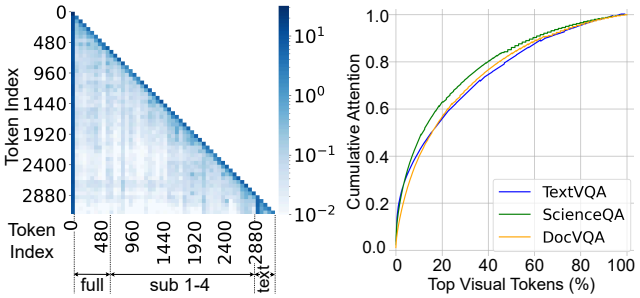
eters. Techniques like model quantization (Dettmers et al. 2022) and weight pruning or masking (Sun et al. 2024) further reduce resource demands but fail to address the critical issue of excessive visual tokens. Token ensemble frameworks such as CrossGET (Shi et al. 2024) and MADTP Cao et al. (2024) primarily target cross-modal transformer-based VLMs, such as BLIP (Li et al. 2023a). Methods like Q-Former, M^3 (Cai et al. 2024), and Abstractor (Cha et al. 2024) require expensive training and fine-tuning.

Sparse Attention Computation in LLM and ViT. These methods aim to reduce the computational cost of attention mechanisms in the transformer layers. FastV (Chen et al. 2025b) identifies important visual tokens in the initial layers of LLM and skips unimportant tokens in subsequent layers, but it is not designed for high-resolution VLMs. While FlexAttention (Li et al. 2025) can handle high-resolution images, it does not allow control over the number of visual tokens based on resource constraints. Methods such as DynamicViT (Rao et al. 2021), PuMer (Cao, Paranjape, and Hajishirzi 2023), and EViT (Tao et al. 2024) are primarily for ViTs, and thus, their efficiency gains are limited as the majority of computation occurs in the LLM.

Early Dropping of Visual Tokens. Methods like TokenCorrCompressor (Zhang et al. 2024c) and PruMerge (Shang et al. 2024) drop visual tokens from the image encoding stage before feeding them to the LLM for greater efficiency. TokenCorrCompressor identifies repetitive whitespace patterns in document images through token-to-token cosine similarity and drops redundant tokens with high similarity. However, their work considers document understanding tasks only. PruMerge prunes out visual tokens with low `CLS`-attention and merges them with selected tokens, and PruMerge+, an enhanced version, selects additional spatial tokens to mitigate information loss. Since PruMerge is designed for low-resolution VLMs, the model accuracy degrades significantly for high-resolution VLMs (see Section 5.1). Moreover, these methods lack control over the number of visual tokens within the memory budget, which is essential in resource-constrained environments.

3 Key Insights

Sparse visual tokens with high attention scores. In a typical VLM, the LLM processes visual, text, and system tokens together. To understand the role of various tokens in the LLM generation phase, we investigate their attention patterns on LLaVA-Next-7B (see Fig. 2a). This experiment reveals that while visual tokens amount to 80-90% of all the



(a) Attention scores of tokens (b) CDF of attention scores

Figure 2: The sparse nature of visual tokens is evident during the generation using LLM. (a) Visual tokens receive significantly less attention compared to system and text tokens. (b) The top 20% and 40% of visual tokens account for 60% and 80% of the total attention, respectively.

tokens, they receive significantly less attention than system and text tokens. To further examine the gap between tokens with high and low attention scores, we compute the Cumulative Distribution Function (CDF) of attention scores for top visual tokens with the highest attention scores (see Fig. 2b). The results reveal that a small subset of visual tokens brings most of the context from the image to the LLM.

Insight 1 (Visual token sparsity) *Despite the large number of visual tokens, only a small subset is important in the LLM generation phase, suggesting an opportunity to drop less important tokens without sacrificing accuracy.*

Various Importance of Sub-images. In Section 1, we discussed how dynamic partitioning can significantly boost accuracy by encoding global and detailed local representations through full-image and sub-images. To understand the contribution of image partitions in the LLM generation phase, we count the number of tokens with the highest attention for each image partition in Table 2. It shows that the distribution of top visual tokens varies across different image partitions.

Insight 2 (Sub-images with different content amounts) *The variation in the visual content weights of image partitions suggests that some partitions may allow more token dropping than others.*

4 Our Design: HiRED

The above observations suggest that only a subset of visual tokens is crucial during the LLM generation phase and the varying importance of image partitions presents a clear opportunity for dropping various numbers of visual tokens from image partitions under a fixed token budget. Motivated by these useful insights, we further explore the `CLS`-attention pattern in ViT (Section 4.1) and propose a novel *attention-guided token-dropping* scheme (Section 4.2).

4.1 `CLS`-attention Pattern in ViT

As discussed in Section 3, it is straightforward to identify important visual tokens (those with higher attention scores)

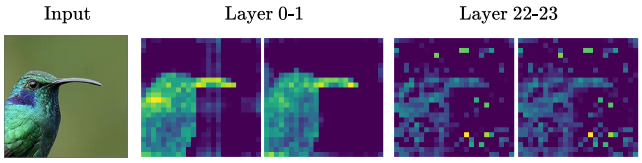


Figure 3: In ViT, CLS-attention map shows distinct characteristics across layers. The initial layers highlight the subject patches while ignoring the background, aligning mostly with the image content. The final layers, however, highlight informative patches where ViT stores most of the image features.

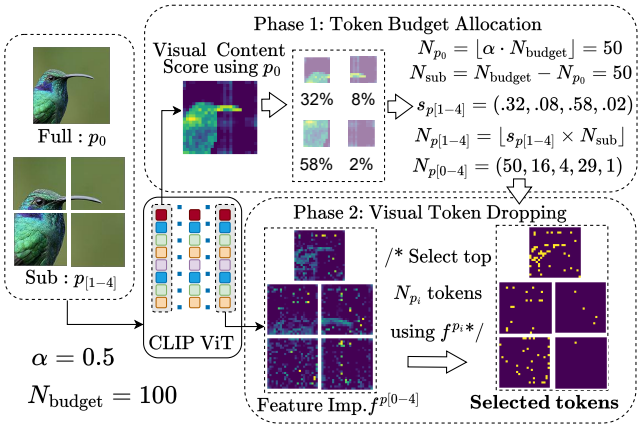


Figure 4: Design of HiRED for high-resolution VLMs to drop visual tokens before LLM. We first allocate token budgets for the full-image and sub-images and then select tokens with top feature importance within the allocated budget.

during the LLM generation phase. However, to reduce input sequence length and improve computational efficiency, it is crucial to identify important tokens earlier, before the generation phase. To do this, we utilize the CLS-attention of ViT. Particularly, CLIP (Radford et al. 2021) splits an input image into a fixed number of non-overlapping patches. In the transformer layers, the CLS token is employed to extract useful information from the patch tokens. By design, the CLS-attention indicates the importance of patches (Gandelsman, Efros, and Steinhardt 2024). Since the patch tokens are later transformed into visual tokens for the subsequent LLM, a guided selection of patches can effectively reduce the number of tokens. Now, a key question is: *How do we identify important visual tokens before the LLM generation phase that are crucial for generating the response?*

To answer this question, we analyze the CLS-attention map across ViT layers and uncover two important findings. First, CLS-attention maps in *initial layers* reveal the main content of the input image. The highlighted patches in these attention maps correspond to visually important parts of the image. As shown in Fig. 3, the attention maps of the first two layers highlight patches derived from *the bird* while ignoring background areas with insignificant visual content. Second, attention maps in *final layers* indicate the informative areas, i.e., patches containing more image features. In

Algorithm 1: HiRED

Input: N_{budget}, N_{ViT}, α, k, l_{init}, l_{final}, H, T_{p_i}, $\{a^{p_i}_{l,h}[j]\}$.

/* Phase 1: Token Budget Allocation */
// 1-1. Compute the visual token budgets N_{p_0} and N_{sub} for the full-image and all the sub-images, respectively
$N_{p_0} \leftarrow \lfloor \alpha \cdot N_{\text{budget}} \rfloor$;
$N_{\text{sub}} \leftarrow N_{\text{budget}} - N_{p_0}$;
// 1-2. Compute budget N_{p_i} for each sub-image partition p_i
// Calculate visual content score s_{p_i} for each sub-image partition p_i using initial layer's CLS-attention of full-image p_0
For each sub-image partition p_i with $i = 1 : k$ **do**
  $s_{p_i} \leftarrow \sum_{j \in T_{p_i}} \sum_{h=1}^{H} a^{p_0}_{l_{\text{init}},h}[j]$;
end For
// Allocate budget N_{p_i} for each sub-image partition p_i
For each sub-image partition p_i with $i = 1 : k$ **do**
  $N_{p_i} \leftarrow \lfloor N_{\text{sub}} \cdot \frac{s_{p_i}}{\sum_{j=1}^{k} s_{p_j}} \rfloor$;
end For

/* Phase 2: Visual Token Dropping */
For each image partition p_i with $i = 0 : k$ **do**
  // 2-1. Compute feature importance score $f^{p_i}[j]$ for each token j using final layer's CLS-attention of image partition p_i
  For $j = 1 : N_{\text{ViT}}$ **do**
    $f^{p_i}[j] \leftarrow \sum_{h=1}^{H} a^{p_i}_{l_{\text{final}},h}[j]$;
  end For
  // 2-2. Select important visual tokens within the budget
  Select top N_{p_i} visual tokens with the highest $f^{p_i}[j]$;
end For

the last two layers, the highlighted patches are distributed across both the image content and the background. As the ViT processes the image, it encodes local features into corresponding patches in the initial layers. However, in the final layers, it learns the relationships between these local features and encodes them into a few background patches as global features (Darcet et al. 2024). As a result, the highlighted areas in the CLS-attention of final layers prioritize patches that are more informative than the others (Pan et al. 2021).

4.2 HiRED Design

Inspired by the above findings on CLS-attention, we propose HiRED, an attention-guided token-dropping scheme comprising two phases: 1) *Token Budget Allocation*, which determines the drop ratio for each image partition, given a total token budget; 2) *Visual Token Dropping*, which selects the most informative visual tokens (and drops the rest) according to the drop ratio determined in Phase 1. The overall design is illustrated in Fig. 4 and detailed in Algorithm 1.

Phase 1: Token Budget Allocation. Let $\{p_0, p_1, \ldots, p_k\}$ denote the set of $(k+1)$ image partitions, where p_0 denotes the full-image and p_i denotes the i-th sub-images. Let T_{p_i} denote the set of token indices of the full-image corresponding to sub-image p_i. Each partition consists of N_{ViT} tokens (e.g., 576 for CLIP), totaling $N_{\text{ViT}} \cdot (k+1)$ tokens before dropping. Given a token budget N_{budget}, we allocate it across $(k+1)$ image partitions and use N_{p_i} to denote the budget of image partition p_i. Consider the ViT consisting of H heads and L layers. Let $\{a^{p_i}_{l,h}[j]\}_{j \in T_{p_i}}$ be the CLS-

Model & Method	Budget	Visual QA		Transcription			Others		
		VQA^{v2}	SQA	VQA^{T}	DocVQA	OCRBench	MME	POPE	ChartQA
LLaVA-Next-7B	Full	80.3	73.2	64.8	73.4	501	1519	87.6	54.8
Spatial	40%	77.7	68.0	57.0	58.9	369	1401	87.2	39.0
PruMerge	10% (Avg.)	75.6	66.8	53.5	37.8	336	1393	85.0	28.8
PruMerge+	55% (Avg.)	78.0	68.2	54.4	44.6	365	1474	87.9	30.2
HiRED	20%	77.5	73.4	61.4	60.8	475	1483	87.0	42.0
HiRED	40%	78.8	73.8	63.6	68.7	488	1474	88.2	46.5
LLaVA-Next-13B	Full	80.9	73.6	66.9	77.5	508	1572	87.1	66.2
Spatial	40%	79.1	73.0	58.8	61.3	390	1529	87.2	42.6
PruMerge	10% (Avg.)	74.1	69.2	54.4	45.9	381	1471	84.9	31.0
PruMerge+	55% (Avg.)	79.1	70.7	55.9	45.9	381	1480	87.5	31.0
HiRED	20%	77.9	71.9	63.6	64.3	462	1545	86.7	48.9
HiRED	40%	79.3	73.2	65.2	72.5	491	1570	87.7	53.7

Table 3: Accuracy comparison between HiRED and the baselines. In all metrics, higher values indicate better performance. Here, VQA^{v2}, SQA, and VQA^{T} stand for VQA-v2, ScienceQA, and TextVQA, respectively.

attention score at layer l and head h for partition p_i, and let l_{init} and l_{final} be the initial and final layers, respectively.

First, we allocate the budget N_{budget} between the full-image and a set of all sub-images using a budget *allocation ratio* $\alpha \in [0, 1]$. The budget for the full-image is $N_{p_0} := \lfloor \alpha \cdot N_{\text{budget}} \rfloor$, and the remaining budget, $N_{\text{sub}} := N_{\text{budget}} - N_{p_0}$, is allocated to the sub-images. Through experiments, we determine the optimal value of α (see Section 5.3).

Second, we distribute the remaining budget N_{sub} across k sub-images. As discussed in Section 4.1, the CLS-attention map of the initial layer on the full-image captures the distribution of visual contents in the image. That is, higher attention corresponds to regions with more visual content, indicating that these regions require a larger token budget (i.e., less dropping). To formalize this, we compute a *visual content score* s_{p_i} for each sub-image p_i (excluding the full-image p_0) as follows:

$$s_{p_i} := \sum\nolimits_{j \in T_{p_i}} \sum\nolimits_{h=1}^{H} a^{p_0}_{l_{\text{init}},h}[j], \forall i \in \{1, 2, \cdots, k\}. \quad (1)$$

Specifically, for each sub-image p_i, we aggregate the CLS-attention across all corresponding tokens on the full-image (i.e., T_{p_i}) and across all H heads of the initial layer ($l_{\text{init}} = 0$). Then, the budget for each sub-image is determined by its fraction of the total: $N_{p_i} := \lfloor N_{\text{sub}} \cdot \frac{s_{p_i}}{\sum_{j=1}^{k} s_{p_j}} \rfloor$. This token budget guides the token dropping in the next phase.

Phase 2: Visual Token Dropping. Our token-dropping scheme aims to retain the most informative visual tokens. To achieve this, we introduce a *feature importance score* for each partition, denoted by $\{f^{p_i}[j]\}_{j \in \{1,2,\cdots,N_{\text{ViT}}\}}$. As observed in Section 4.1, the CLS-attention map of the final layer highlights the informative patches from both an image partition's subject and background areas. Moreover, different heads learn different features (Gandelsman, Efros, and Steinhardt 2024). We thus compute the feature importance score $f^{p_i}[j]$ for the j-th token in each partition p_i as follows:

$$f^{p_i}[j] := \sum\nolimits_{h=1}^{H} a^{p_i}_{l_{\text{final}},h}[j], \quad (2)$$

for all $i \in \{0, 1, \cdots, k\}$ and $j \in \{1, 2, \cdots, N_{\text{ViT}}\}$. Specifically, we add CLS-attention of the final layer ($l_{\text{final}} = 22$ across all heads. We make these design choices based on the experiments on the impact of different layers and head aggregation strategies (see Section 5.3).

Finally, we select the N_{p_i} number of tokens with the highest feature importance score $f^{p_i}[j]$ and drop the rest of the tokens for each partition p_i. The selected visual tokens (along with text and system tokens) are then concatenated and fed into the subsequent LLM.

5 Evaluation

We evaluate HiRED on LLaVA-Next (Liu et al. 2024a), LLaVA-v1.5 (Liu et al. 2023), and ShareGPT4V (Chen et al. 2025a). For performance evaluation, we use an entry-level NVIDIA TESLA P40 (24 GB) GPU.

Downstream Tasks and Benchmarks. We used eight benchmarks from LMMS-EVAL (Zhang et al. 2024b) evaluation framework across three different task types: 1) *Visual Question Answering (VQA)* includes high-level object recognition benchmarks such as VQA-v2 (Goyal et al. 2017) and ScienceQA (Lu et al. 2022); 2) *Transcription* focuses on fine-grained transcription tasks, including TextVQA (Singh et al. 2019), DocVQA (Mathew, Karatzas, and Jawahar 2021), and OCRBench (Liu et al. 2024b); and 3) *Others* consists of MME (Fu et al. 2024) for perception and cognition abilities, POPE (Li et al. 2023b) for hallucination detection and ChartQA (Masry et al. 2022) for spatial understanding.

Baselines. We select PruMerge and PruMerge+ (Shang et al. 2024) as our primary baselines because they utilize early-dropping mechanisms similar to ours. PruMerge uses spatial redundancy in visual tokens and performs pruning and merging on visual tokens, and PruMerge+, an enhanced version, additionally includes visual tokens through spatially uniform sampling to minimize accuracy losses. Since PruMerge and PruMerge+ are designed for LLaVA (a low-resolution VLM without dynamic partitioning), we apply

Batch Size	Budget	Throughput (tokens/sec)	Latency (sec)	Memory (GB)
1	Full	0.49	19.49	16.04
	40%	1.40	6.91	14.33
	20%	2.30	4.21	13.76
2	Full	0.66	44.37	21.84
	40%	2.04	14.31	16.81
	20%	3.68	7.89	15.07
4	Full	–	–	OOM
	40%	2.22	26.38	20.36
	20%	4.28	13.60	16.99

Table 4: Inference efficiency: throughput, latency, and GPU memory usage across different batch sizes using LLaVA-Next-7B with HiRED under various token budgets.

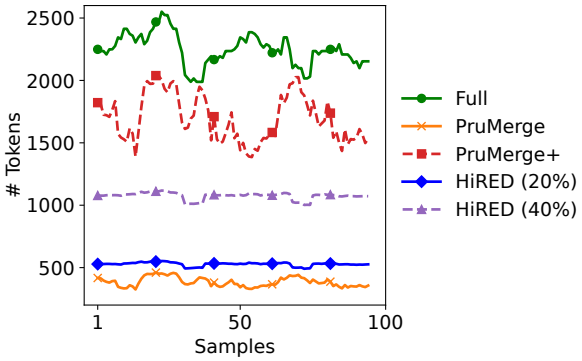


Figure 5: Number of visual tokens generated in 100 samples of TextVQA for Full, PruMerge, PruMerge+, and HiRED

their token-dropping strategy to each image partition. Additionally, we include spatial pooling as a simple baseline suggested in previous works (Marin et al. 2023). While TokenCorrCompressor (Li et al. 2025) also supports early token dropping, they consider document understanding tasks only, and the code was not publicly available at the time of writing, rendering a direct comparison infeasible. Although FastV (Chen et al. 2025b) and FlexAttention (Li et al. 2025) optimize VLMs through sparse attention in LLMs, they require processing excessive visual tokens in the earlier layers of LLMs, leading to inefficiencies in both latency and memory compared to early-dropping methods.

5.1 Accuracy

We evaluate the accuracy of HiRED on LLaVA-Next (7B and 13B) and compare it with the baselines for various tasks. The results are presented in Table 3. To study the robustness of HiRED, we further evaluate low-resolution VLMs (with a single partition) such as LLaVA-1.5-7B and ShareGPT4V. The accuracy metrics reported in the table are the default metrics used for the corresponding tasks.

Accuracy vs. Token Reduction. Evaluation results show that with a 20% token budget (i.e., a maximum of 576 tokens), HiRED achieves nearly the same accuracy as full execution (i.e., a maximum of 2880 tokens) for VQA tasks. With a 40% token budget (i.e., a maximum of 1152 tokens), it maintains comparable accuracy for fine-grained transcription tasks. Interestingly, for ScienceQA and POPE, we observe an increase in accuracy with fewer tokens. This suggests that in some cases, reducing the number of tokens to some extent may even improve accuracy.

Comparison with Baselines. We observe a greater accuracy degradation across all tasks for both PruMerge and PruMerge+. While these methods can dynamically adjust the token budget to retain more visual information when necessary, they still fall short compared to HiRED, particularly in transcription tasks. On average, PruMerge and PruMerge+ use 10% and 55% of tokens, respectively, for transcription tasks (see Section 5.2). However, PruMerge+, even with 55% of tokens (on average), has 11% and 26% lower accuracy than HiRED-20% for TextVQA and DocVQA, respectively. Similarly, PruMerge with 10% tokens, has 13% and 37% lower accuracy, respectively.

5.2 Inference Efficiency

To evaluate the inference efficiency of HiRED, we measure: 1) the number of visual tokens; 2) token generation throughput; 3) time-to-first-token (TTFT) latency; 4) GPU memory usage. The results are presented in Table 4. A comparison of token usage with the baselines is illustrated in Fig. 5.

Inference Efficiency of HiRED. Table 4 highlights the inference efficiency of HiRED. With a 20% token budget, HiRED achieves a 4.7× increase in token generation throughput (2.30 vs. 0.49 tokens/sec) compared to the full execution (i.e., 100% tokens). It also reduces the TTFT latency by 78% (4.21 vs. 19.49 seconds), which is crucial for low-latency applications. Furthermore, HiRED-20% reduces GPU memory usage by 14% (13.76 vs. 16.04 GB). For larger batch sizes, it even shows higher gain. For instance, with a batch size of 4, full execution encounters out-of-memory (OOM) on the 24 GB GPU. In contrast, HiRED-20% reduces memory usage by 30% (16.99 GB) while maintaining the throughput and latency improvements.

Efficiency under Token Budget. Fig. 5 demonstrates that HiRED's token usage in LLaVA-Next-7B consistently remains within the predefined resource constraints (e.g., 20%). In contrast, full execution without dropping (Full), PruMerge, and PruMerge+ exhibit significant variations in the number of visual tokens across different samples in TextVQA. For Full, the variation arises from differences in partitioning based on the width-to-height ratio and resolution of image as well as the removal of some padding tokens. For PruMerge and PruMerge+, the variation stems from their adaptive nature, which allocates more tokens to images with higher visual information density. This fluctuation in the number of visual tokens directly affects computational costs, while HiRED enforces a strict token budget, making it well-suited in resource constraints.

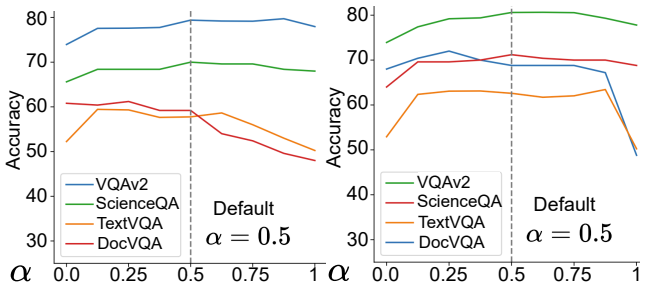


Figure 6: Accuracy vs. Budget Allocation Ratio α for token budget of 20% (left) and 40% (right).

Choice	SQA	VQA^{v2}	VQA^{T}	DocVQA
Distribute budget:				
evenly	67.6	73.3	52.2	48.8
using layer 22	65.2	68.6	37.3	54.0
using layer 0	**68.4**	**77.7**	**54.8**	**59.2**
Drop tokens:				
using layer 0	65.2	68.6	37.3	54.0
using layer 11	68.0	76.1	51.9	52.0
using layer 22	**68.4**	**77.7**	**54.8**	**59.2**
Aggregate heads:				
No agg	67.2	76.6	50.5	52.4
Addition	**68.4**	**77.7**	**54.8**	**59.2**

Table 5: Ablation study of ViT layer selection and head aggregation strategy in HiRED's token dropping algorithm.

5.3 Ablation Study

We evaluate the key components of HiRED and the effectiveness of our design choices. For this study, we select two VQA tasks (i.e., ScienceQA and VQA-v2) and two fine-grained transcription tasks (i.e., TextVQA and DocVQA).

Value of Budget Allocation Ratio α. Fig. 6 shows that budget allocation ratio α can impact the accuracy. Specifically, choosing $\alpha = 0$ indicates no token budget allocated to the full-image, while $\alpha = 1$ assigns the entire token budget to the full-image. The remaining token budget is distributed among the sub-images based on their importance. This result highlights that balancing the budget between the full-image and sub-images is crucial, and a balanced budget distribution (i.e., $\alpha = 0.5$) generally yields the highest accuracy. Therefore, we choose $\alpha = 0.5$ as the default value for allocating the token budget between the full-image and sub-images.

Design Choices. We evaluate the design choices of HiRED in Table 5. We use the `CLS`-attention from the initial ViT layer ($l_{\text{init}} = 0$) to allocate the token budget. This choice is motivated by the stronger alignment of early-layer attentions with the visual content of the input image, compared to deeper layers like Layer 22. Even when the budget is distributed evenly across image partitions (e.g., dividing a budget of 100 among 5 partitions as 20 each), using Layer 0 achieves the best results. We determine the drop ratios using Layer 0 for token dropping but perform the actual dropping based on Layer 22. Dropping tokens using other layers (e.g., Layer 0 or Layer 11) leads to lower performance, as deeper layers in ViTs aggregate information into fewer, more informative tokens. Thus, using the final layer helps identify the most critical tokens. Furthermore, our head aggregation strategy, which combines attention scores through summation, achieves higher accuracy compared to no aggregation.

Model	Budget	SQA	VQA^{v2}	VQA^{T}	DocVQA
LLaVA-1.5-7B	Full	69.5	76.6	46.1	28.1
HiRED	40%	67.2	79.0	47.0	29.4
HiRED	20%	66.4	74.7	44.2	24.6
ShareGPT4V-7B	Full	68.4	80.6	50.7	26.76
HiRED	40%	67.2	79.0	50.0	25.97
HiRED	20%	66.4	74.7	49.3	24.76

Table 6: Accuracy comparison on low-resolution (i.e., single partition) using LLaVA-1.5-7B and ShareGPT4V-7B.

Low-Resolution VLMs. We further evaluate HiRED for two low-resolution VLMs (i.e., models without dynamic partitioning) such as LLaVA-1.5-7B and ShareGPT4V-7B. This evaluation serves two purposes: 1) to demonstrate the robustness and wide applicability of HiRED across different VLMs; 2) to isolate the performance of the token-dropping strategy by excluding budget allocation, which does not apply to single-partition inputs. As shown in Table 6, HiRED maintains accuracy close to full execution for LLaVA-1.5-7B and ShareGPT4V-7B, even with a limited token budget (40% and 20%). This result highlights the robustness and effectiveness of our token-dropping scheme.

6 Conclusion

High-resolution VLMs significantly enhance multimodal capability by retaining detailed image information, but the excessive number of visual tokens poses significant challenges during inference. To address this challenge, we proposed HiRED, a plug-and-play token-dropping framework that allocates a fixed token budget across image partitions, prioritizes the most informative visual tokens, and drops the rest before LLM generation. Our evaluations demonstrate that HiRED substantially improves inference throughput, reduces latency, and lowers GPU memory consumption while maintaining competitive accuracy across diverse multimodal benchmarks. We believe HiRED provides a practical and scalable solution for deploying high-resolution VLMs in resource-constrained environments and offers a foundation for further optimization of multimodal inference systems.

A limitation of HiRED is the potential loss of spatial information, which may impact tasks where spatial relationships are crucial, such as task understanding in ChartQA. This limitation arises as LLMs rely on positional encodings optimized for language modeling, which are less suited for visual tokens. One possible solution is to incorporate 2D positional encodings after token dropping to preserve spatial relationships, which we leave for future work.

Acknowledgments

This work was supported in part by the National Science Foundation under Grants CNS-2315851 and 2106634, the Commonwealth Cyber Initiative, a Sony Faculty Innovation Award under AG3ZURVF, the Department for the Economy, Northern Ireland, under grant agreement USI-226, and Science Foundation Ireland under grant agreement 22/US/3848.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2024. GPT-4 Technical Report. arXiv:2303.08774.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond. arXiv:2308.12966.

Cai, M.; Yang, J.; Gao, J.; and Lee, Y. J. 2024. Matryoshka Multimodal Models. arXiv:2405.17430.

Cao, J.; Ye, P.; Li, S.; Yu, C.; Tang, Y.; Lu, J.; and Chen, T. 2024. MADTP: Multimodal Alignment-Guided Dynamic Token Pruning for Accelerating Vision-Language Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15710–15719.

Cao, Q.; Paranjape, B.; and Hajishirzi, H. 2023. PuMer: Pruning and Merging Tokens for Efficient Vision Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12890–12903. Toronto, Canada: Association for Computational Linguistics (ACL).

Cha, J.; Kang, W.; Mun, J.; and Roh, B. 2024. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13817–13827.

Chen, L.; Li, J.; Dong, X.; Zhang, P.; He, C.; Wang, J.; Zhao, F.; and Lin, D. 2025a. Sharegpt4v: Improving large multimodal models with better captions. In *European Conference on Computer Vision (ECCV)*, 370–387.

Chen, L.; Zhao, H.; Liu, T.; Bai, S.; Lin, J.; Zhou, C.; and Chang, B. 2025b. An Image is Worth 1/2 Tokens After Layer 2: Plug-and-Play Inference Acceleration for Large Vision-Language Models. In Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; and Varol, G., eds., *Computer Vision – ECCV 2024*, 19–35. Cham: Springer Nature Switzerland. ISBN 978-3-031-73004-7.

Chen, Z.; Pekis, A.; and Brown, K. 2024. Advancing High Resolution Vision-Language Models in Biomedicine. arXiv:2406.09454.

Chu, X.; Qiao, L.; Lin, X.; Xu, S.; Yang, Y.; Hu, Y.; Wei, F.; Zhang, X.; Zhang, B.; Wei, X.; and Shen, C. 2023. MobileVLM : A Fast, Strong and Open Vision Language Assistant for Mobile Devices. arXiv:2312.16886.

Darcet, T.; Oquab, M.; Mairal, J.; and Bojanowski, P. 2024. Vision Transformers Need Registers. In *The Twelfth International Conference on Learning Representations*.

Dettmers, T.; Lewis, M.; Belkada, Y.; and Zettlemoyer, L. 2022. GPT3.int8(): 8-bit Matrix Multiplication for Transformers at Scale. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 30318–30332. Curran Associates, Inc.

Dong, X.; Zhang, P.; Zang, Y.; Cao, Y.; Wang, B.; Ouyang, L.; Zhang, S.; Duan, H.; Zhang, W.; Li, Y.; Yan, H.; Gao, Y.; Chen, Z.; xinyue zhang; Li, W.; Jingwen, L.; Wang, W.; Chen, K.; He, C.; ZHANG, X.; Dai, J.; Qiao, Y.; Lin, D.; and Wang, J. 2024. InternLM-XComposer2-4KHD: A Pioneering Large Vision-Language Model Handling Resolutions from 336 Pixels to 4K HD. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Fu, C.; Chen, P.; Shen, Y.; Qin, Y.; Zhang, M.; Lin, X.; Yang, J.; Zheng, X.; Li, K.; Sun, X.; Wu, Y.; and Ji, R. 2024. MME: A Comprehensive Evaluation Benchmark for Multimodal Large Language Models. arXiv:2306.13394.

Gandelsman, Y.; Efros, A. A.; and Steinhardt, J. 2024. Interpreting CLIP's Image Representation via Text-Based Decomposition. In *The Twelfth International Conference on Learning Representations*.

Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 6904–6913.

Hu, A.; Xu, H.; Ye, J.; Yan, M.; Zhang, L.; Zhang, B.; Zhang, J.; Jin, Q.; Huang, F.; and Zhou, J. 2024. mPLUG-DocOwl 1.5: Unified Structure Learning for OCR-free Document Understanding. In Al-Onaizan, Y.; Bansal, M.; and Chen, Y.-N., eds., *Findings of the Association for Computational Linguistics: EMNLP 2024*, 3096–3120. Miami, Florida, USA: Association for Computational Linguistics.

Li, J.; Chen, D.; Cai, T.; Chen, P.; Hong, Y.; Chen, Z.; Shen, Y.; and Gan, C. 2025. FlexAttention for Efficient High-Resolution Vision-Language Models. In Leonardis, A.; Ricci, E.; Roth, S.; Russakovsky, O.; Sattler, T.; and Varol, G., eds., *Computer Vision – ECCV 2024*, 286–302. Cham: Springer Nature Switzerland. ISBN 978-3-031-72698-9.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning (ICML)*, 19730–19742. PMLR.

Li, Y.; Du, Y.; Zhou, K.; Wang, J.; Zhao, X.; and Wen, J.-R. 2023b. Evaluating Object Hallucination in Large Vision-Language Models. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 292–305. Singapore: Association for Computational Linguistics.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26763–26773.

Lin, Z.; Liu, C.; Zhang, R.; Gao, P.; Qiu, L.; Xiao, H.; Qiu, H.; Lin, C.; Shao, W.; Chen, K.; Han, J.; Huang, S.;

Zhang, Y.; He, X.; Li, H.; and Qiao, Y. 2023. SPHINX: The Joint Mixing of Weights, Tasks, and Visual Embeddings for Multi-modal Large Language Models. arXiv:2311.07575.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. 2024a. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. In *NeurIPS*.

Liu, Y.; Li, Z.; Huang, M.; Yang, B.; Yu, W.; Li, C.; Yin, X.-C.; Liu, C.-L.; Jin, L.; and Bai, X. 2024b. OCRBench: on the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12).

Liu, Y.; Yang, B.; Liu, Q.; Li, Z.; Ma, Z.; Zhang, S.; and Bai, X. 2024c. TextMonkey: An OCR-Free Large Multimodal Model for Understanding Document. arXiv:2403.04473.

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Marin, D.; Chang, J.-H. R.; Ranjan, A.; Prabhu, A.; Rastegari, M.; and Tuzel, O. 2023. Token pooling in vision transformers for image classification. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 12–21.

Masry, A.; Do, X. L.; Tan, J. Q.; Joty, S.; and Hoque, E. 2022. ChartQA: A Benchmark for Question Answering about Charts with Visual and Logical Reasoning. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Findings of the Association for Computational Linguistics: ACL 2022*, 2263–2279. Dublin, Ireland: Association for Computational Linguistics.

Mathew, M.; Karatzas, D.; and Jawahar, C. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2200–2209.

McKinzie, B.; Gan, Z.; Fauconnier, J.-P.; Dodge, S.; Zhang, B.; Dufter, P.; Shah, D.; Du, X.; Peng, F.; Belyi, A.; et al. 2025. MM1: methods, analysis and insights from multimodal LLM pre-training. In *European Conference on Computer Vision*, 304–323. Springer.

Pan, B.; Panda, R.; Jiang, Y.; Wang, Z.; Feris, R.; and Oliva, A. 2021. IA-RED^2: Interpretability-Aware Redundancy Reduction for Vision Transformers. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 24898–24911. Curran Associates, Inc.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; and Hsieh, C.-J. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34: 13937–13949.

Reid, M.; Savinov, N.; Teplyashin, D.; Lepikhin, D.; Lillicrap, T.; Alayrac, J.-b.; Soricut, R.; Lazaridou, A.; Firat, O.; Schrittwieser, J.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. arXiv:2403.05530.

Shang, Y.; Cai, M.; Xu, B.; Lee, Y. J.; and Yan, Y. 2024. LLaVA-PruMerge: Adaptive Token Reduction for Efficient Large Multimodal Models. arXiv:2403.15388.

Shi, D.; Tao, C.; Rao, A.; Yang, Z.; Yuan, C.; and Wang, J. 2024. CrossGET: Cross-Guided Ensemble of Tokens for Accelerating Vision-Language Transformers. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, 44960–44990. PMLR.

Singh, A.; Natarajan, V.; Shah, M.; Jiang, Y.; Chen, X.; Batra, D.; Parikh, D.; and Rohrbach, M. 2019. Towards VQA Models That Can Read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Sun, M.; Liu, Z.; Bair, A.; and Kolter, J. Z. 2024. A Simple and Effective Pruning Approach for Large Language Models. In *The Twelfth International Conference on Learning Representations*.

Tao, Z.; Chen, X.; Jin, Z.; Bai, X.; Zhao, H.; and Lou, Y. 2024. EVIT: Event-Oriented Instruction Tuning for Event Reasoning. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 8966–8979. Bangkok, Thailand: Association for Computational Linguistics.

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Zhang, D.; Yu, Y.; Dong, J.; Li, C.; Su, D.; Chu, C.; and Yu, D. 2024a. MM-LLMs: Recent Advances in MultiModal Large Language Models. In Ku, L.-W.; Martins, A.; and Srikumar, V., eds., *Findings of the Association for Computational Linguistics: ACL 2024*, 12401–12430. Bangkok, Thailand: Association for Computational Linguistics.

Zhang, K.; Li, B.; Zhang, P.; Pu, F.; Cahyono, J. A.; Hu, K.; Liu, S.; Zhang, Y.; Yang, J.; Li, C.; and Liu, Z. 2024b. LMMs-Eval: Reality Check on the Evaluation of Large Multimodal Models. arXiv:2407.12772.

Zhang, R.; Lyu, Y.; Shao, R.; Chen, G.; Guan, W.; and Nie, L. 2024c. Token-level Correlation-guided Compression for Efficient Multimodal Document Understanding. arXiv:2407.14439.

Zhou, B.; Hu, Y.; Weng, X.; Jia, J.; Luo, J.; Liu, X.; Wu, J.; and Huang, L. 2024. TinyLLaVA: A Framework of Small-scale Large Multimodal Models. arXiv:2402.14289.

Zhu, Y.; Zhu, M.; Liu, N.; Xu, Z.; and Peng, Y. 2024. LLaVA-Phi: Efficient Multi-Modal Assistant with Small Language Model. In *Proceedings of the 1st International Workshop on Efficient Multimedia Computing under Limited*, EMCLR'24, 18–22. New York, NY, USA: Association for Computing Machinery. ISBN 9798400711909.