This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Mostly Text, Smart Visuals: Asymmetric Text-Visual Pruning for Large Vision-Language Models

Sijie Li
University of Sheffield, UK
sli256@sheffield.ac.uk

Biao Qian
Tsinghua University, China
hfutqian@gmail.com

Jungong Han*
Tsinghua University, China
jghan@tsinghua.edu.cn

Abstract

*Network pruning is an effective technique for enabling lightweight Large Vision-Language Models (LVLMs), which primarily incorporates both weights and activations into the importance metric. However, existing efforts typically process calibration data from different modalities in a unified manner, overlooking modality-specific behaviors. This raises a critical challenge: how to address the divergent behaviors of textual and visual tokens for accurate pruning of LVLMs. To this end, we systematically investigate the sensitivity of visual and textual tokens to the pruning operation by decoupling their corresponding weights, revealing that: (i) the textual pathway should be calibrated via text tokens, since it exhibits higher sensitivity than the visual pathway; (ii) the visual pathway exhibits high redundancy, permitting even 50% sparsity. Motivated by these insights, we propose a simple yet effective Asymmetric Text-Visual Weight Pruning method for LVLMs, dubbed **ATV-Pruning**, which establishes the importance metric for accurate weight pruning by selecting the informative tokens from both textual and visual pathways. Specifically, ATV-Pruning integrates two primary innovations: first, a calibration pool is adaptively constructed by drawing on all textual tokens and a subset of visual tokens; second, we devise a layer-adaptive selection strategy to yield important visual tokens. Finally, extensive experiments across standard multimodal benchmarks verify the superiority of our ATV-Pruning over state-of-the-art methods. Code is available at https://github.com/LezJ/ATV-Pruning.*

1. Introduction

Large Vision-Language Models (LVLMs) have exhibited outstanding capabilities in various tasks, such as visual question answering [1, 20, 21], embodied intelligence [10, 39], and video understanding [6, 35]. Despite the remarkable performance, LVLMs are typically parameter-heavy and computationally demanding, which pose critical challenges for deployment on resource-constrained devices, *e.g.*, robots and autonomous vehicles.

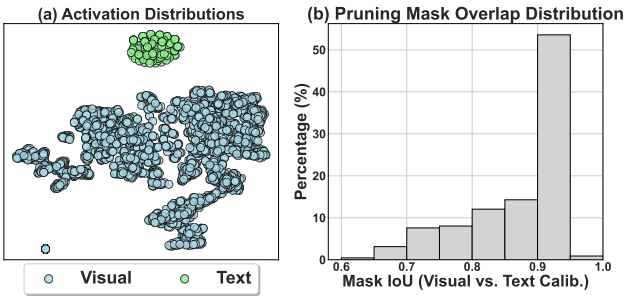


Figure 1. **Illustration of the divergent statistical characteristics across different modalities**, manifesting as: (a) *activation representation*: the activations induced by textual and visual inputs to the LVLM occupy distinct clustered regions in the representation space (t-SNE visualization); and (b) *pruning importance*: the pruning masks across all linear layers, derived from the text-only and visual-only calibration data, exhibit a broad IoU distribution (taking 50% sparsity level as an example).

To remedy this issue, weight pruning has emerged as a powerful technique to remove redundant model parameters. In the unimodal Large Language Models (LLMs) domain, post-training methods like SparseGPT [11] and efficient activation-aware approaches [8, 28, 33] (e.g., Wanda [28]) have achieved remarkable success with a small calibration set, i.e., a held-out set of representative samples used to estimate pruning criteria. Nevertheless, naively applying these LLMs-centric techniques to LVLMs generally suffers from suboptimal performance, since different input modalities can exhibit distinct sensitivity to the pruned sub-networks, harming cross-modal alignment and semantic integrity. Recently, several LVLM-specific studies have proposed to seek optimal block-wise sparsity allocations [19, 29] and explore modality-aware schemes [16]. For instance, TAMP [16] uses multimodal statistics to guide calibration token selection, but still treats visual and textual tokens within a shared selection process. These approaches tend to leverage mixed-modal activations in a largely uniform manner for importance assessment, hence they still fail to address the persistent issue of modality heterogeneity.

*Corresponding author

We identify the divergent statistical characteristics across different modalities as two aspects: *activation representation*, where the activations induced by textual and visual inputs to the LVLM occupy distinct clustered regions in the representation space (see Fig. 1(a)); and *pruning importance*, where the pruning masks derived from the text-only and visual-only calibration data across all linear layers of the LVLM exhibit a broad IoU distribution (see Fig. 1(b)), indicating that the weights deemed important by the textual modality can differ significantly from those identified by the visual modality. The above motivates us to delve into *how different modalities behave differently under pruning operations* and *how to effectively perform activation-aware pruning in LVLMs*.

To answer these questions, we conduct a series of motivation experiments that decouple text-associated and visual-associated parameters within Transformer blocks and independently study the sensitivity of different input modalities. Our motivation experiments reveal the following findings:

(i) The textual pathway is highly sensitive to the selection of the calibration pool. Pruning the textual pathway with visual-only or mixed calibration pool substantially degrades the accuracy. For instance, on SQA_{img}, the model performance collapses to 35.85 and 11.1 at 60% sparsity level with mixed or visual-only calibration pool; in contrast, text-only calibration pool reaches 61.58 (see Tab. 1).

(ii) The visual pathway is highly redundant and robust. Pruning the visual pathway to 60% sparsity preserves over 99.25% performance across all calibration pools, *even with text-only calibration* (Tab. 1), suggesting that text calibration already identifies most visually crucial weights, with only a few visual tokens needed to complement the text pool and capture the remaining visual-specific parameters.

Motivated by these observations, in this paper, we propose a novel Asymmetric Text-Visual Weight Pruning method for LVLMs, named **ATV-Pruning**, which explicitly accounts for the inherent differences between textual and visual modalities during pruning. Specifically, ATV-Pruning comprises two key components: (i) a cross-modal calibration pool is constructed by adaptively integrating all textual tokens with a sparse subset of visual tokens; to achieve this, (ii) a block-adaptive selection strategy is devised to identify and retain the most crucial visual tokens. The resulting pool is then used to evaluate the weight importance scores for weight pruning of LVLMs. Extensive experiments across standard multimodal benchmarks and models demonstrate the merits of ATV-Pruning over state-of-the-art methods.

Our major contributions are summarized as follows:

- We provide novel insights into the asymmetric pruning sensitivity across modalities: (i) text-side pruning is highly sensitive and must be calibrated with text tokens, while (ii) the visual pathway is highly robust, admitting 60% unstructured sparsity with minimal degradation.
- We propose ATV-Pruning, a simple yet effective pruning framework that adaptively constructs a cross-modal calibration pool by integrating full textual tokens with a compact, layer-adaptive subset of visual tokens.
- Comprehensive evaluation on nine standard multimodal benchmarks demonstrate the superiority of ATV-Pruning over state-of-the-art baselines.

2. Related Work

LVLMs and efficiency. The remarkable success of Large Language Models (LLMs) [3, 25, 30] in text-based reasoning has spurred the development of Large Vision-Language Models (LVLMs) [1, 20, 21], which extend these capabilities into the visual domain. Early LVLMs demonstrated strong performance on foundational tasks like visual question answering [1, 20]. Recently, the field has rapidly advanced towards more complex and data-intensive capabilities. For instance, LLaVA-NeXT [21] pushes performance in fine-grained reasoning and OCR by processing images at significantly higher resolutions. Simultaneously, models like LLaVA-OneVision [17] and Qwen2-VL [32] have been developed as unified systems that can process single-image, multi-image, and video inputs to perform more complex reasoning. While impressive, these advancements directly exacerbate the computational challenges. They are expensive to serve in practice: the LLM backbone remains large, and supporting high-resolution images or long video sequences introduces exceptionally long visual token sequences. Consequently, the inference latency and memory cost scale accordingly, making deployment on resource-constrained systems a critical bottleneck.

Several directions have been explored to improve efficiency. One line aims to shrink model size directly via smaller backbones [7, 38]. Another line applies quantization [12, 31], reducing precision of weights and/or activations for faster inference. A third line targets sequence redundancy through visual token pruning [2, 5, 37], which drops a large fraction of vision tokens with minimal loss. Our work is orthogonal: we focus on *weight pruning* to sparsify the shared backbone and improve efficiency. We additionally show that redundancy is not only in visual tokens but also in the visual-processing pathway parameters, which can tolerate surprisingly high sparsity.

Weight pruning methods. For LLMs, SparseGPT [11] prunes weights by approximately minimizing output error using a local Hessian-based update. This yields strong performance but requires an iterative weight update procedure per layer, which adds pruning overhead. Wanda [28] instead proposes an activation-aware importance score based on weight magnitude and input activation norms; it achieves competitive retention without expensive weight updates, making it attractive for fast post-hoc sparsification.

Extending pruning to LVLMs, prior work has primarily examined how to allocate sparsity budgets across Transformer blocks to account for the multimodal nature [16, 19, 29], whereas our work focuses on improving the pruning criterion itself. Among these methods, TAMP [16] further introduces Adaptive Multimodal Input Activation: it selects a subset of highly attended, diverse text and visual tokens as calibration input for Wanda scoring. While this improves performance over vanilla Wanda, the selection remains heuristic and introduces significant extra compute. Crucially, the modality-specific pruning behavior is not explicitly examined. In contrast, we start from a modality-disentangled analysis that reveals the distinct sensitivities of text and visual pathways under different calibration pools; guided by these findings, ATV-Pruning applies an asymmetric token-selection strategy—keeping all text tokens while layer-adaptively sampling a small set of informative visual tokens—to obtain more stable pruning.

3. Methodology

Our goal is to prune the LLM backbone of an LVLM under high unstructured sparsity, while preserving both linguistic capability and multimodal reasoning quality. We argue that, in multimodal settings, *the choice of calibration tokens used to estimate activation statistics is critical*, as activation distributions differ substantially across modalities. In this section, we begin by revisiting the activation-aware pruning scheme (Sec. 3.1), represented by Wanda [28]. Following that, we perform the motivation experiments to offer insights into the weight pruning for LVLMs (Sec. 3.2). Building on this, we elaborate our proposed Asymmetric Text-Visual Weight Pruning method (Sec. 3.3).

3.1. Preliminaries

Our pruning recipe is built on top of *Wanda* [28], a post-hoc, activation-aware method, which evaluates the weight importance scores by combining weight magnitude and estimated input activation.

Weight importance scoring. Consider a linear layer with weight matrix $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ and corresponding input activations $\mathbf{X} \in \mathbb{R}^{N \times d_{\text{in}}}$, where N indexes token positions across a calibration batch. An importance score is assigned to each scalar weight $\mathbf{W}_{ij}$ via

$$\mathbf{I}_{ij} = |\mathbf{W}_{ij}| \cdot \|\mathbf{X}_j\|_2, \tag{1}$$

where $\mathbf{X}_j \in \mathbb{R}^N$ denotes the j-th input channel across all tokens in the calibration batch, and $\|\cdot\|_2$ is the Euclidean norm. Intuitively, a weight is considered important when it is *both large in magnitude and frequently activated*. With $\mathbf{I}_{ij}$ for all entries in $\mathbf{W}$, we can prune the lowest-scoring $\rho\%$ weights, yielding unstructured sparsity of the layer.

Standard calibration for LLMs. In unimodal LLM pruning, $\|\mathbf{X}_j\|_2$ in Eq. 1 is computed over *all* tokens from a small text-only calibration corpus. This implicitly assumes: (i) all inference-time tokens "behave similarly enough" to share statistics, and (ii) the model is dominated by a single modality, so the calibration pool is homogeneous.

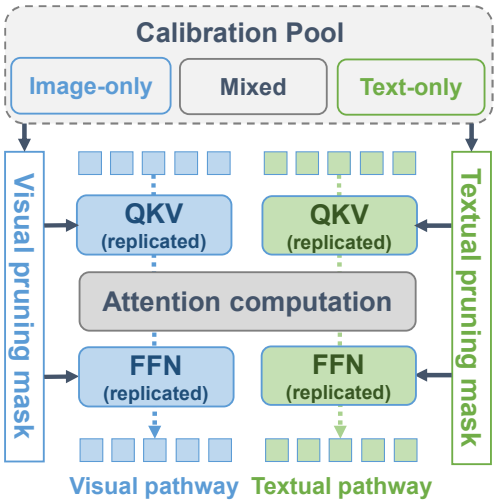


Figure 2. **Modality decoupling via MoT probe.** For each Transformer block, the QKV and FFN layers are **replicated into visual and textual pathways**, which process their respective token types. Independent pruning masks are derived for each pathway using activation statistics from *text-only*, *image-only*, or *mixed* calibration pools. This setup enables controlled comparison of modality-specific pruning sensitivity.

Modality-Agnostic Calibration Pool for LVLMs. Unlike LLMs, a typical input sequence for LVLMs interleaves text tokens with visual tokens. When the weight pruning shifts from LLMs to LVLMs, the naive strategy is to apply the principles of Wanda to LVLMs, which reuses Eq. 1 and compute $\|\mathbf{X}_j\|_2$ over a calibration pool that comprises all multimodal tokens (i.e., both text and visual tokens). We refer to this "use everything" approach as the *modality-agnostic calibration pool*. It is straightforward to implement and remains a common practical choice in LVLM pruning pipelines. Although several recent studies have begun to consider the multimodal nature of calibration data [16, 19, 29], a systematic investigation of how different calibration sources (text-only, image-only, or mixed) influence pruning behavior across text- and image-associated weights has been lacking. To bridge this gap, we conduct a controlled, *modality-aware* sensitivity analysis as discussed in the next.

3.2. Motivation Investigation: Modality-Aware Sensitivity Analysis

To provide insights into the effects of textual and visual modalities on the weight pruning process, we aim to delve into the following critical question: *for the activation-aware pruning scheme in LVLMs, which type of tokens should be exploited to estimate activation statistics while preserv-*

Pathway	Calibration	MMB	SQA_{img}	VizWiz	Average
Dense		72.16	73.28	57.65	100%
50% Sparsity					
Text	Text	65.03	70.55	62.94	98.26%
	Visual	61.77	67.67	60.56	94.33%
	Mixed	64.00	69.01	60.36	95.86%
Visual	Text	71.39	74.17	58.03	100.27%
	Visual	70.87	73.13	57.71	99.37%
	Mixed	71.56	73.92	57.87	100.14%
60% Sparsity					
Text	Text	52.15	61.58	56.30	84.65%
	Visual	41.58	11.11	46.11	50.92%
	Mixed	44.76	35.85	48.41	64.97%
Visual	Text	71.05	73.67	58.32	100.05%
	Visual	70.53	73.08	57.81	99.25%
	Mixed	70.88	73.38	57.85	99.57%

Table 1. **Sensitivity analysis of both modality pathways** across different calibration token sources, benchmarks, and sparsity.

ing overall performance? Therefore, we design a series of motivation experiments, where the textual and visual pathways are explicitly disentangled without harming the original computation flows, with LLaVA-NeXT [21] (more experimental details in the Supplementary). Specifically, we introduce a Mixture-of-Transformer (MoT) [9, 26] analysis probe (Fig. 2), where (i) the shared weights for each transformer block are replicated and allocated to the textual and visual pathway; (ii) the textual and visual pathway are independently pruned under varied calibration pools. In particular, we select either of the textual, visual or both type tokens to compute $\|\mathbf{X}_j\|_2$, which is further utilized to evaluate the weight importance scores I_{ij} in Eq. 1. According to the type of selected tokens, the calibration pool involves the following three cases: **text-only calibration pool** to use only positions corresponding to text tokens; **image-only calibration pool** to use only positions corresponding to visual tokens; **mixed calibration pool** to use all tokens from the full multimodal sequence. With the above settings, Tab. 1 reports experimental results under the 2×3 design at 50% and 60% sparsity, and reveals the following critical findings.

- ***Finding A: Asymmetry calibration sensitivity across textual and visual pathways.*** For the *textual pathway* pruning, the best retention is consistently achieved when $\|\mathbf{X}_j\|_2$ in Eq. 1 is computed using *text-only* calibration pool. Meanwhile, using mixed calibration pool behaves better than image-only calibration pool. In contrast, for *visual pathway* pruning, using different calibration pools receives the similar pruning performance, hence the visual pathway is less sensitive to the type of calibration pool compared to the textual pathway. Such fact indicates that text-anchored calibration is necessary to safeguard linguistic and reasoning capability, while using text-driven calibration generally works well even when pruning weights processing visual tokens.
- ***Finding B: The visual pathway possesses significant weight-level redundancy.*** We observe that, when applying 50% or 60% unstructured sparsity solely to the visual pathway, the pruning performance is comparable to the dense baseline, pointing to substantial over-parameterization in visual processing. Combined with *Finding A*, the facts suggest that, during calibration, it is neither necessary nor beneficial to treat all visual tokens as equally informative. Instead, a compact subset of salient visual evidence is sufficient to protect vision performance. Note that this redundancy differs from token-level redundancy explored by visual token pruning studies [2, 5, 37], where the former lies in parameters within the backbone, the latter in input token sequence.

Interestingly, on VizWiz at 50% sparsity, we observe a performance gain over the dense model—an effect also noted in prior LVLM pruning study [19], suggesting that removing ambiguous neurons can improve performance on certain benchmarks. Our analysis extends this finding by showing that such beneficial pruning mainly occurs within the textual pathway, especially when applying text-only calibration. This further reinforces that text calibration is essential, and sometimes even beneficial for LVLM pruning.

Takeaway. A modality-agnostic calibration pool dilutes the linguistic signal essential for protecting text-associated weights. Calibration should be text-anchored and augmented only with a compact set of salient visual tokens.

3.3. Asymmetric Text-Visual Pruning

Motivated by the findings in Sec. 3.2, we propose an Asymmetric Text-Visual Weight Pruning (ATV-Pruning) method for LVLMs, which performs modality-aware weight pruning by considering the sensitivity difference of textual and visual pathways towards calibration source. We first discuss how to construct a modality-aware calibration set.

3.3.1. Modality-Aware Calibration Set

The above investigation conveys two principles for constructing an effective calibration set: textual tokens are essential for preserving language competency; and a subset of visual tokens is sufficient to maintain robust vision ability. Building on that, we replace the conventional "use everything" calibration pool with a *modality-aware calibration set* that (i) includes *all text tokens* and (ii) selecting a *small, layer-adaptive subset of high-impact visual tokens*. Formally, for each calibration sample, let $\mathcal{T}$ be the set of all text-token positions, and let $\mathcal{V}_{\text{sub}}$ be a block-adaptive subset of salient visual-token positions. For each transformer block, the modality-aware calibration set is defined as

$$\mathcal{S}_{\text{cal}} = \mathcal{T} \cup \mathcal{V}_{\text{sub}} \tag{2}$$

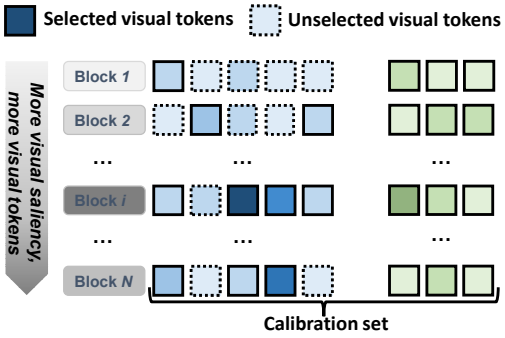


Figure 3. **Overview of ATV-Pruning;** Color intensity reflects the degree of visual saliency. Blocks with higher visual saliency keep more salient visual tokens, with all text tokens.

This set is then used to evaluate the importance score according to Eq. 1. Next, we discuss the critical question of how to perform block-adaptive visual selection.

3.3.2. Block-Adaptive Visual Selection

Visual token saliency in pruning. We posit that visual tokens do not contribute equally to optimal calibration statistics, and their computational role varies significantly across transformer blocks. Motivated by this, we present a block-adaptive visual selection strategy guided by *token saliency*. Specifically, given a Transformer block b and a visual token v from the calibration set, we define its saliency score as

$$s_v \;=\; \phi_b(v), \tag{3}$$

where $\phi_b(\cdot)$ is a block-specific scoring function that evaluates the importance of v at block b.

Instantiating saliency via visual drift. While various choices of ϕ_b are possible, we find that a token's *representation drift* within a block is a robust proxy for its saliency. Intuitively, if a block substantially updates a visual token, that token is actively involved in the computation and should be prioritized during calibration. Let $\mathbf{X}_{\text{in},v}$ and $\mathbf{X}_{\text{out},v}$ denote the input and output representations of token v at block b. We instantiate ϕ_b as the cosine distance between these representations:

$$s_v \;=\; 1 - \cos\big(\mathbf{X}_{\text{in},v},\, \mathbf{X}_{\text{out},v}\big), \tag{4}$$

where larger s_v indicates stronger visual drift and thus higher saliency.

From saliency to block-wise budgets. To capture how visually active a block is on average, we aggregate token-level saliency scores over all visual tokens observed at that block across the calibration set:

$$\bar{s} \;=\; \frac{1}{|\mathcal{V}_{\text{all}}|} \sum_{v \in \mathcal{V}_{\text{all}}} s_v, \tag{5}$$

where $\mathcal{V}_{\text{all}}$ denotes the multiset of all visual tokens at that block across the calibration batch. Empirically, $\bar{s}$ is small in early layers and larger in mid-to-late layers, reflecting the progressive strengthening of visual information processing. We then allocate a visual-token budget K to each block as

$$K \;=\; \Big\lfloor \alpha \cdot \bar{s} \cdot n_{\text{text}} \Big\rfloor, \tag{6}$$

where n_{text} is the number of text tokens for the current calibration sample and $\alpha > 0$ is a global scaling hyperparameter. Blocks with higher average visual saliency (larger $\bar{s}$) are thus allowed to retain more visual tokens, leading to a *block-adaptive* allocation of visual-token budgets.

Visual token selection given budgets. Given the per-block budget K, we perform saliency-based token selection within each calibration sample. For each sample, let $\mathcal{V}$ be the set of its visual-token indices at block b and $\{s_v : v \in \mathcal{V}\}$ the corresponding saliency scores. We keep only the top-K most salient visual tokens:

$$\mathcal{V}_{\text{sub}} \;=\; \text{TopK}_v\big(\{s_v\},\, K\big). \tag{7}$$

We then form the modality-aware calibration set for that block via $\mathcal{S}_{\text{cal}} = \mathcal{T} \cup \mathcal{V}_{\text{sub}}$ (Eq. 2), and use only the positions in $\mathcal{S}_{\text{cal}}$ to estimate activation norms $\|\mathbf{X}_j\|_2$ in Eq. 1.

By construction, the resulting importance scores are *text-anchored* and *visually selective*, directing pruning towards weights that matter most for LVLMs' capabilities.

Alternative saliency signals. Although visual drift, *i.e.*, cosine distance change (Eq. 4), is our default saliency signal, we also experimented with alternative choices inspired by visual token pruning studies, such as attention received

Algorithm 1 ATV-Pruning procedure

Require: Model weights; calibration set $\mathcal{D}_{\text{cal}}$; scaling factor α; target sparsity ρ.

```
for each Transformer block do
    V_all ← all visual tokens in D_cal at this block
    Compute s_v for all v ∈ V_all using Eq. 4
    Compute s̄ with s_v via Eq. 5
    for each calibration sample do
        T ← text-token indices in the sample
        n_text ← |T|
        V ← visual-token indices in the sample
        Compute K = ⌊α · d̄ · n_text⌋ (Eq. 6)
        Select V_sub from V using Eq. 7
        Form S_cal = T ∪ V_sub (Eq. 2)
    end for
    Pool all S_cal across calibration samples
    Estimate ‖X_j‖_2 only from pooled S_cal positions
    Compute I_ij = |W_ij| · ‖X_j‖_2 (Eq. 1)
    Prune the lowest ρ% of weights according to I_ij
end for
```

Ensure: Pruned model $\{\mathbf{W}_{\text{pruned}}\}$.

from text queries [5, 37] and diversity-oriented token selection [2]. These alternatives can induce similar block-adaptive behavior, but visual drift yields the most stable performance (see the validation experiments in Sec.4.3.2).

3.3.3. Optimization Pipeline

The above specifies our asymmetric text-visual weight pruning framework, as illustrated in Fig. 3. Specifically, for each Transformer block, we kick off ATV-Pruning by performing a forward pass with the calibration set to compute the per-token saliency s_v and the block-average $\bar{s}$. The budget K is then derived through the global scaling factor α, and the modality-aware calibration set $\mathcal{S}_{\text{cal}}$ is formed by combining all text tokens with the top-K silent visual tokens for later activation-aware pruning. This pipeline ensures that the pruning metric is dominated by linguistic signals while retaining critical visual context. The whole pruning procedure is summarized in Algo. 1

4. Experiments

4.1. Experimental Settings and Details

Evaluation benchmarks. To evaluate our ATV-Pruning, we conduct the experiments on nine standard multimodal benchmarks, including: GQA [15], MMBench-EN (MMB) [22], MME (perception and cognition) [13], MMMU [34], OK-VQA [24], POPE [18], ScienceQA-IMG (SQA_{img}) [23], TextVQA [27], and VizWiz-VQA [14]. A brief introduction to each benchmark is provided in the Supplementary. We evaluate the model performance via two metrics: the official metric provided by each benchmark, and *average performance retention* across all benchmarks, computed as the mean value of the pruned model's performance divided by that of its dense counterpart.

Implementation details. Our primary experiments are conducted using LLaVA-NeXT (8B) [21] at 50% and 60% unstructured sparsity levels. Unless otherwise specified, we fix the global scaling factor at $\alpha = 1.0$ when pruning LLaVA-NeXT, which yields on average 15 selected visual tokens per sample per block over the calibration pool. In addition, we evaluate Qwen2-VL (7B) [32] at 60% sparsity level, and report the experimental results with $\alpha = 1.5$. For fair comparison, we use the same calibration samples across all models and sparsity levels. Following TAMP [16], the calibration set contains 128 high-quality image–text pairs randomly sampled from ShareGPT4V [4]. Across all methods, we apply *uniformly distributed unstructured sparsity* to the weight matrices of all linear layers in the Attention and MLP blocks of the LLM backbone. The compared methods, *i.e.*, SparseGPT [11], Wanda [28], and TAMP [16], are reproduced using their official configurations. All experiments run on a single NVIDIA A100 GPU, and we employ lmms-eval [36] to ensure a unified evaluation protocol.

4.2. Comparison with State-of-the-art Approaches

To validate the superiority of our ATV-Pruning, we compare it with the typical pruning approaches for LLMs and LVLMs, including: SparseGPT [11] approximates output preservation through local weight updates; Wanda [28] incorporates both weights and activations into the importance evaluation; TAMP [16] performs modality-aware pruning via a multimodal-informed threshold. Table 2 reports the performance of the pruned models under unstructured, uniform pruning and summarizes the following findings.

- ***LLaVA-NeXT:*** At 50% sparsity, ATV-Pruning achieves the *highest average retention* of ***94.00%***, ranking first on GQA, MME, MMMU, OK-VQA, and VizWiz, while remaining competitive on the other benchmarks. At 60% sparsity, our method again yields the *best average* at ***77.01%***. Despite SparseGPT often being strong at higher sparsity in LLMs with its computationally intensive weight update process [28], ATV-Pruning tops MMMU, OK-VQA, SQA_{img}, and VizWiz. The advantage over other Wanda-style baselines widens as sparsity increases: at 60%, ATV-Pruning exceeds Wanda and TAMP by ***+12.56*** and ***+3.26*** average points, respectively. Across both pruning levels, ATV-Pruning matches or outperforms these baselines on most tasks.
- ***Qwen2-VL:*** We further validate the generalizability of our method on Qwen2-VL 7B. At 60% sparsity, ATV-Pruning maintains its lead with the highest average retention of ***85.65%***. It outperforms SparseGPT and TAMP by +0.35 and +1.86 points, respectively, while securing the ***best results on 6 of 9*** benchmarks, proving its stability..

Overall, across both model architectures and pruning levels, ATV-Pruning consistently matches or outperforms state-of-the-art baselines on the majority of tasks, with remarkable pruning efficiency (see Sec. 4.5 for details). We provide additional results for other models and N:M semi-structured sparsity in the Supplementary.

4.3. Ablation Studies

We conduct controlled ablations to understand each component constituting our ATV-Pruning. Unless specified otherwise, all experiments are performed at 50% unstructured sparsity level with LLaVA-NeXT (8B) backbone, and evaluated on MMB [22], SQA_{img} [23], and VizWiz [14].

4.3.1. Effect of Scaling Factor α

The global scaling factor α in Eq. 6 determines the block-level visual-token budget K, which reflects how many salient visual tokens are incorporated into the calibration pool. To validate its effectiveness, we perform the ablation experiments by setting varied $\alpha \in \{0, 0.5, 1, 2, 4, 8\}$. Fig. 4 illustrates that ATV-Pruning achieves competitive results even under the extreme case $\alpha = 0$, where ATV-Pruning holds a text-only calibration pool. For instance,

Methods	GQA	MMB	MME	MMMU	OKVQA	POPE	SQA_{img}	TextVQA	VizWiz	Average
LLaVA-NeXT 8B, Dense	65.34	72.16	1965.12	40.11	60.13	87.84	73.28	65.42	57.65	100%
LLaVA-NeXT 8B	*50% Sparsity*									
SparseGPT [11]	62.86	**65.38**	1742.80	36.00	38.17	**89.23**	68.12	**64.21**	60.12	91.74%
Wanda [28]	62.61	63.83	1620.02	35.56	25.38	88.02	69.06	64.04	60.73	88.36%
TAMP [16]	63.15	64.86	1732.54	34.78	43.59	88.04	**70.00**	62.78	62.50	92.67%
ATV-Pruning (Ours)	**63.28**	65.29	**1801.51**	**36.11**	**45.34**	88.22	69.86	62.76	**63.34**	**94.00%**
LLaVA-NeXT 8B	*60% Sparsity*									
SparseGPT [11]	**58.52**	**50.60**	**1586.34**	29.33	5.56	**88.58**	55.97	**61.06**	53.50	76.24%
Wanda [28]	49.88	43.38	1107.72	27.78	0.90	82.68	38.67	54.37	49.85	64.45%
TAMP [16]	54.62	48.37	1385.71	30.00	6.73	88.40	55.78	56.36	54.04	73.75%
ATV-Pruning (Ours)	57.61	50.00	1474.65	**31.67**	**11.13**	88.32	**59.30**	55.83	**55.52**	**77.01%**
Qwen2-VL 7B, Dense	62.35	78.95	2306.50	50.89	52.12	88.56	84.98	82.02	68.33	100%
Qwen2-VL 7B	*60% Sparsity*									
SparseGPT [11]	56.98	71.13	1621.23	41.00	**29.50**	88.11	78.53	**78.51**	62.28	85.30%
Wanda [28]	54.37	69.42	1637.66	37.22	6.30	88.38	76.85	75.76	58.84	77.78%
TAMP [16]	57.47	70.02	**1923.30**	41.22	15.93	88.40	77.49	77.24	63.65	83.79%
ATV-Pruning (Ours)	**58.13**	**71.82**	1816.56	**41.56**	23.73	**88.90**	**79.08**	76.96	**63.82**	**85.65%**

Table 2. **Comparison results with state-of-the-art approaches on nine benchmarks under unstructured, uniform pruning.** The official metric for each benchmark and average performance retention (denoted as Average) across all benchmarks, are reported to evaluate the performance of pruned models. Best results are highlighted in **green**

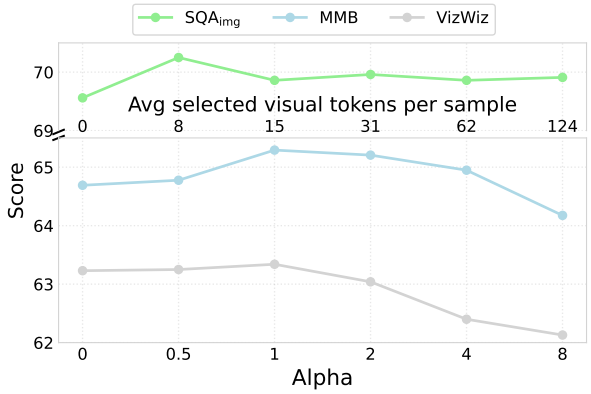


Figure 4. **Ablation study about global scaling factor α.** Larger α allocates more salient visual tokens. A moderate range $\alpha \in [0.5, 2]$ yields stable performance across benchmarks.

it reaches 64.69 on MMB, outperforming naive Wanda (63.83)— *highlighting the critical role of text tokens in providing stable activation statistics*. Notably, introducing a moderate amount of visual information with $\alpha \in [0.5, 2]$ can improve multimodal robustness, yielding consistently high retention across benchmarks. Thereby, to avoid excessive tuning, we fix $\alpha = 1$ for all LLaVA-NeXT experiments and report results on this setting across all benchmarks in our main experiments. Beyond this range, *i.e.*, $\alpha \in \{4, 8\}$, the performance of ATV-Pruning begins to degrade as redundant visual tokens are incorporated into the calibration pool, which dilutes the linguistic signal and drives the results closer to the modality-agnostic baseline. The above fact indicates that *maintaining a text-anchored calibration while selectively adding a compact set of high-impact visual tokens strikes superior accuracy.*

Signal	Adaptive	MMB	SQA_{img}	VizWiz	Average
Dense		72.16	73.28	57.65	100%
Random	✗	64.52	69.26	62.59	97.50%
Drift	✗	64.95	69.56	63.30	98.24%
Our ATV-Pruning					
Drift	✓	**65.29**	69.86	**63.34**	**98.56%**
ABS	✓	65.12	**70.65**	62.59	98.41%
DBS	✓	64.78	70.10	62.84	98.14%

Table 3. **Discussion on selection strategy of visual tokens.** All methods select the same total number of visual tokens for calibration. "Adaptive" indicates whether the visual-token budget is dynamically distributed across blocks. Our default configuration achieves the best overall retention.

4.3.2. Discussion on Selection Strategy of Visual Tokens

In the main experiments, we default to using *visual drift*, *i.e.*, the cosine similarity between a token's input and output within a block (Eq. 4), together with *adaptive visual-token budget allocation* across blocks (Sec. 3.3.2). To verify the effectiveness of such setting, we perform the ablation experiments from the following two cases: replacing adaptive saliency-based selection with fixed random selection, denote as "Random, ✗"; removing adaptivity and fixing the per-block budget, denote as "Drift, ✗", where all variants are compared under a controlled setting and the *total number of selected visual tokens is kept identical* to ensure fairness. As shown in Tab. 3, our ATV-Pruning achieves 98.56% average retention. In comparison, "Drift, ✗" drops to 98.24%, and the "Random, ✗" drops further to 97.50%.

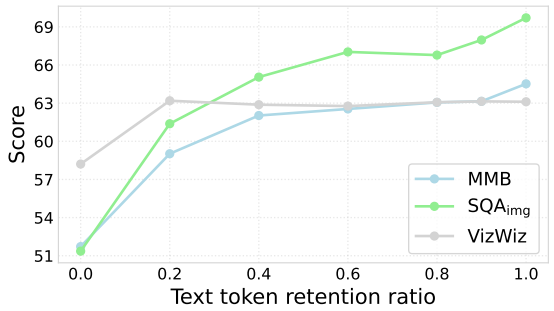


Figure 5. **Effect of textual token selection under varied retained ratios during calibration**, with the default visual token subset of ATV-Pruning. Removing text tokens consistently harms the performance, highlighting their necessity for stable pruning.

This implies that (*i*) a meaningful saliency signal (drift) surfaces more informative visual tokens than random selection, and (*ii*) distributing the visual-token budget according to block activity further achieves stable performance.

Additionally, we discuss two alternative visual saliency metrics drawn from recent visual token pruning studies, *i.e.*, *attention-based signal*, denoted as ABS, which measures how much attention each visual token receives from text queries; and *diversity-based strategy*, denoted as DBS, which encourages selecting visually diverse tokens. Implementation details of both are in the Supplementary. Both ABS and DBS benefit from adaptive allocation and deliver reasonable results, as reported in Tab. 3. For instance, ABS achieves the highest SQA_{img} score of 70.65. However, our ATV-Pruning's adaptive visual drift achieves the best balance across benchmarks and the highest average retention, confirming that per-block representational change serves as an effective and lightweight cue for identifying salient visual tokens during pruning.

4.4. Further Insights into Textual Token Selection

Recalling that all textual tokens are included for the calibration pool in Sec.3.3, we further discuss whether textual tokens should be adaptively selected during calibration, similar to visual tokens. To investigate this, we progressively reduce the proportion of text tokens used to compute activation norms, with similar drift saliency, while keeping the default visual token selection of ATV-Pruning. We conduct the experiments with varied retained text-token ratio (*i.e.*, $\{1.0,0.9,0.8,0.6,0.4,0.2,0\}$). Fig. 5 illustrates that *not selecting all text tokens always leads to performance degradation of ATV-Pruning*. Crucially, for more difficult benchmarks, such as MMB (comprehensive multimodal ability) and SQA_{img} (science-oriented reasoning), even reducing the retained text-token ratio from 1.0 to 0.9 leads to a noticeable performance drop. For example, MMB decreases from 64.52 to 63.14, while SQA_{img} drops from 69.71 to 67.97, suggesting that even the least-changed text tokens may still contribute to preserving linguistic grounding and cross-modal reasoning. Unlike visual tokens—where selective inclusion improves robustness—the text pathway is not redundant. The above *confirms that full text-token retention for ATV-Pruning provides the most reliable activation statistics for pruning.*

Method	Time (s)↓	Rel. time↓	Gain (pp)↑
Wanda [28]	73.7	1.00×	+0
SparseGPT [11]	666.0	9.04×	+3.38
TAMP [16]	1418.4	19.25×	+4.31
ATV-Pruning (Ours)	99.6	1.35×	+5.64

Table 4. **Pruning efficiency analysis**, on LLaVA-NeXT (8B) at 50% sparsity. Time (s): wall-clock pruning time in seconds; Rel. time: pruning time relative to Wanda; Gain (pp): average performance gain in percentage points (pp), taken from Tab. 2.

4.5. Pruning Efficiency Analysis

To investigate the practical efficiency of ATV-Pruning, we conduct the experiments on LLaVA-NeXT (8B) by measuring the end-to-end wall-clock time to achieve 50% unstructured sparsity under a unified pruning protocol. The results from Table 4 suggest that ATV-Pruning (99.6s) exhibits obvious efficiency advantages over SparseGPT (666.0s) and TAMP (1418.4s). The reasons are that SparseGPT introduces a compute-intensive weight update procedure, while TAMP involves additional attention computation and an iterative maximum mean discrepancy (MMD)-based token selection process. Notably, compared to Wanda that merely performs a single forward pass to collect activation norms, ATV-Pruning introduces a modest computational overhead (1.35×), while achieving substantial performance advantages (*+5.64* percentage points). In particular, the extra time cost of ATV-Pruning primarily stems from the computation of the cosine-distance-based visual drift $\bar{s}$ in Eq. 4. Such fact indicates that ATV-Pruning is capable of serving as a simple, plug-in module for LVLMs pruning.

5. Conclusion

In this paper, we introduced ATV-Pruning, a simple yet effective method for pruning LVLMs that explicitly accounts for their asymmetric modal sensitivities. Our approach is based on the core insight that the textual pathway is highly sensitive, while the robust visual pathway possesses significant redundancy. ATV-Pruning leverages this by constructing an asymmetric calibration pool, using all textual tokens but only a sparse, block-adaptively selected subset of visual tokens. Experimental results demonstrate that ATV-Pruning achieves state-of-the-art performance, outperforming prior methods with high pruning efficiency. Future work will focus on extending this asymmetric pruning paradigm to Large Multimodal Models (LMMs), which incorporate a wider range of modalities, including audio, depth, etc.

Acknowledgements

This work was supported in part by Beijing Natural Science Foundation (L257005), and is also supported by National Natural Science Foundation of China No. 62441235.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 1, 2

[2] Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9392–9401, 2025. 2, 4, 6

[3] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023. 2

[4] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Conference on Computer Vision*, pages 370–387. Springer, 2024. 6

[5] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024. 2, 4, 6

[6] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 1

[7] Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, strong and open vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023. 2

[8] Rocktim Jyoti Das, Mingjie Sun, Liqun Ma, and Zhiqiang Shen. Beyond size: How gradients shape pruning decisions in large language models. *arXiv preprint arXiv:2311.04902*, 2023. 1

[9] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025. 4

[10] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: an embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, pages 8469–8488, 2023. 1

[11] Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pages 10323–10337. PMLR, 2023. 1, 2, 6, 7, 8

[12] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022. 2

[13] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 6

[14] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018. 6

[15] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 6

[16] Jaewoo Lee, Keyang Xuan, Chanakya Ekbote, Sandeep Polisetty, Yi R. Fung, and Paul Pu Liang. TAMP: Token-adaptive layerwise pruning in multimodal large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6892–6908, Vienna, Austria, 2025. Association for Computational Linguistics. 1, 3, 6, 7, 8

[17] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2

[18] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 6

[19] Yinan Liang, Ziwei Wang, Xiuwei Xu, Jie Zhou, and Jiwen Lu. Efficientllava: Generalizable auto-pruning for large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9445–9454, 2025. 1, 3, 4

[20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 1, 2

[21] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024. 1, 2, 4, 6

[22] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024. 6

[23] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and

Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022. 6

[24] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 3195–3204, 2019. 6

[25] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 2

[26] Weijia Shi, Xiaochuang Han, Chunting Zhou, Weixin Liang, Xi Victoria Lin, Luke Zettlemoyer, and Lili Yu. Lmfusion: Adapting pretrained language models for multimodal generation. *arXiv preprint arXiv:2412.15188*, 2024. 4

[27] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 6

[28] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 2, 3, 6, 7, 8

[29] Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. Ecoflap: Efficient coarse-to-fine layer-wise pruning for vision-language models. In *The Twelfth International Conference on Learning Representations*, 2024. 1, 3

[30] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[31] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Q-vlm: Post-training quantization for large vision-language models. *Advances in Neural Information Processing Systems*, 37:114553–114573, 2024. 2

[32] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2, 6

[33] Yifan Yang, Kai Zhen, Bhavana Ganesh, Aram Galstyan, Goeric Huybrechts, Markus Müller, Jonas M Kübler, Rupak Vignesh Swaminathan, Athanasios Mouchtaris, Sravan Babu Bodapati, et al. Wanda++: Pruning large language models via regional gradients. *arXiv preprint arXiv:2503.04992*, 2025. 1

[34] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024. 6

[35] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553, 2023. 1

[36] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 881–916, 2025. 6

[37] Wangbo Zhao, Yizeng Han, Jiasheng Tang, Zhikai Li, Yibing Song, Kai Wang, Zhangyang Wang, and Yang You. A stitch in time saves nine: Small vlm is a precise guidance for accelerating large vlms. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19814–19824, 2025. 2, 4, 6

[38] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024. 2

[39] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023. 1