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

Supplementary Material

A1. Details of Benchmarks

We describe the details of nine benchmarks used in "***Sec. 3.2. Motivation Investigation: Modality-Aware Sensitivity Analysis***" and "***Sec. 4. Experiments***", including:

- **GQA** [9] evaluates compositional visual reasoning over real-world images. It uses scene graphs from images to generate complex, multi-step questions, aiming to move beyond simple object recognition and reduce the biases found in earlier VQA datasets.
- **MMBench-EN** (referred as **MMB**) [14] is the English portion of MMBench, a bilingual multiple-choice benchmark designed for holistic evaluation of large vision–language models. It covers diverse abilities such as perception, reasoning, math, and world knowledge, with carefully curated questions and quality control to enable fair, scalable comparison across models.
- **MME** [7] is a comprehensive evaluation suite for multimodal large language models that explicitly separates perceptual and cognitive skills. It spans 14 subtasks (e.g., existence, OCR, counting, commonsense, code reasoning, etc), using manually designed instruction–answer pairs to reduce data leakage. *We report the sum score of their perception and cognition (all 14 subtasks) evaluation*.
- **MMMU** [22] is a challenging benchmark featuring questions from college-level problems across six core disciplines, such as Art & Design, Science, Health & Medicine, etc. It is designed to evaluate a model's expert-level knowledge and ability to perform deliberate reasoning with complex, multi-modal information.
- **OK-VQA** [16] is a visual question answering benchmark where answering requires external world knowledge beyond what is directly visible in the image.
- **POPE** [12] is a targeted benchmark for measuring object hallucination in large vision–language models. It builds yes/no questions about the presence of candidate objects in images to quantify hallucination via accuracy and related metrics.
- **ScienceQA-IMG** (referred as $\textbf{SQA}_{\textbf{img}}$) [15] is the image-based subset of ScienceQA, a multimodal science question answering benchmark collected from elementary and high school science curricula, emphasizing the chain of thought (CoT) reasoning ability.
- **TextVQA** [19] focuses on answering questions that require reading and understanding text embedded in images.
- **VizWiz** [8] is a VQA dataset constructed from real visual questions asked by blind and visually impaired users, who capture images with a mobile phone. It features challenging, often low-quality images, making it a realistic benchmark for evaluating practical VQA systems.

A2. Beyond Unstructured Pruning: Additional Discussion on Semi-Structured Pruning

While the main paper focuses on the flexible unstructured pruning to demonstrate the fundamental capability of ATV-Pruning, here we complement those results by evaluating our approach under ***hardware-friendly semi-structured sparsity in the general N:M setting***. This format imposes fixed, non-negotiable structural constraints crucial for enabling efficient inference acceleration on modern GPUs. In the $N{:}M$ setting, in each group of M consecutive weights exactly N remain non-zero. Specifically, we study the NVIDIA-compatible 2:4 pattern and the more flexible 4:8 variant, both yielding uniformly distributed 50% sparsity across linear layers. The 2:4 format, in particular, is natively supported by NVIDIA Ampere and Hopper architectures and allows direct use of sparse Tensor Cores [17]; practical acceleration for LLM backbones (e.g., LLaMA [21]) commonly falls in the range of $1.2\times$–$1.6\times$ [20].

Tab. A1 reports results on LLaVA-NeXT (8B) [13], where we apply $\alpha = 1.0$ for ATV-Pruning in both patterns. As expected, semi-structured pruning is more restrictive than unstructured sparsification, causing all methods to experience noticeable degradation relative to the dense model. Nonetheless, ATV-Pruning consistently shows the strongest robustness: it achieves the highest average retained performance of ***76.29%*** under 2:4 and ***85.54%*** under 4:8, outperforming all baselines by a clear margin and leading on most benchmarks. Notably, the gains persist across both the stricter 2:4 constraint and the more relaxed 4:8 setting, indicating that our pruning criterion reliably preserves essential weights even under rigid structural patterns. These results highlight ATV-Pruning as a practical and effective solution for accelerating LLMs on widely used hardware backends.

A3. An Extension of "*Tab. 2*": Additional Results on Other Backbone Models

To further examine the robustness and generality of our approach, we extend ATV-Pruning to two additional LVLMs: LLaVA-OneVision (7B) [11] and Qwen2.5-VL (7B) [2], with $\alpha = 1.0$. The comparison results at 60% unstructured sparsity are reported in Table A2, which serves as ***an extension of "Tab. 2"***. Overall, ATV-Pruning continues to deliver

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	*2:4 Semi-Structured Sparsity*									
SparseGPT [6]	55.13	42.61	1249.34	28.89	16.79	**88.21**	34.71	**58.58**	53.88	70.86%
Wanda [20]	54.67	37.80	1370.97	28.00	11.07	87.52	**58.08**	55.84	55.01	72.62%
TAMP [10]	56.31	42.44	1401.91	28.67	13.84	87.46	57.76	56.11	57.15	74.90%
ATV-Pruning (Ours)	**57.14**	**46.82**	**1466.78**	**29.56**	**17.91**	87.11	52.16	55.71	**58.01**	**76.29%**
LLaVA-NeXT 8B	*4:8 Semi-Structured Sparsity*									
SparseGPT [6]	59.72	57.39	1451.30	30.44	26.47	87.87	59.59	**62.25**	58.76	82.57%
Wanda [20]	60.01	59.88	1507.97	30.67	16.60	**88.33**	63.91	60.13	56.69	81.52%
TAMP [10]	**60.57**	59.19	1590.90	32.56	20.66	88.20	**64.65**	60.08	57.90	83.57%
ATV-Pruning (Ours)	60.38	**60.57**	**1688.91**	**32.89**	**27.23**	88.01	61.68	60.44	**59.70**	**85.54%**

Table A1. **Semi-structured (*2:4; 4:8*) pruning comparisons** on LLaVA-NeXT (8B). Best results are highlighted in **green**.

Methods	GQA	MMB	MME	MMMU	OKVQA	POPE	SQA_{img}	TextVQA	VizWiz	Average
LLaVA-OneVision 7B, Dense	62.25	80.84	1998.62	49.22	61.00	89.13	95.93	76.05	60.38	100%
LLaVA-OneVision 7B	*60% Sparsity*									
SparseGPT [6]	**58.56**	73.71	**1538.61**	37.33	44.65	89.72	79.03	**72.62**	**62.72**	**88.19%**
Wanda [20]	54.86	72.34	1293.20	37.89	25.28	89.54	78.58	69.64	61.00	78.25%
TAMP [10]	57.64	73.20	1571.36	39.67	37.65	89.54	78.58	69.64	61.00	86.56%
ATV-Pruning (Ours)	58.05	**74.57**	1406.75	**40.11**	**46.15**	**89.78**	**80.47**	70.23	61.98	88.07%
Qwen2.5-VL 7B, Dense	60.49	83.25	2333.14	50.78	42.22	87.48	-	82.93	70.48	100%
Qwen2.5-VL 7B	*60% Sparsity*									
SparseGPT [6]	54.26	72.94	**2165.54**	33.89	**26.52**	88.58	-	76.63	**65.28**	85.75%
Wanda [20]	51.30	73.88	1557.82	34.78	15.80	**89.16**	-	75.60	60.53	78.15%
TAMP [10]	55.76	74.23	1974.18	39.22	22.81	88.52	-	77.96	63.58	85.33%
ATV-Pruning (Ours)	**57.58**	**74.91**	1977.43	**39.89**	23.57	88.56	-	**78.07**	64.73	**86.44%**

Table A2. **Additional pruning results for LLaVA-OneVision and Qwen2.5-VL** at 60% unstructured sparsity, serving as an extension of "Table 2". Best results are highlighted in **green**. SQA_{img} results for Qwen2.5-VL are omitted because the current `lmms-eval` setup does not yield reliable evaluations.

strong performance across architectures, confirming its effectiveness beyond the models discussed in the main paper.

Across both models, ATV-Pruning consistently outperforms Wanda-style baselines (Wanda [20] and TAMP [10]) by a clear margin. For instance, on LLaVA-OneVision, it improves average retention by ***+9.82*** and ***+1.51*** points over Wanda and TAMP, respectively; on Qwen2.5-VL, the gains are ***+8.29*** and ***+1.11***. This underscores that ATV-Pruning better preserves multimodal activation statistics during calibration compared to other activation-aware methods.

Notably, ATV-Pruning is also *competitive with, and often superior to,* SparseGPT [6]. It delivers the best results on a greater number of benchmarks across both models, closely tracking SparseGPT's average performance on LLaVA-OneVision (88.07% vs. 88.19%) while achieving the highest overall average on Qwen2.5-VL with ***86.44%***—a ***+0.69*** point improvement over SparseGPT.

In summary, across all four tested models (LLaVA-NeXT, Qwen2-VL, LLaVA-OneVision, and Qwen2.5-VL) at high sparsity (60%), ATV-Pruning consistently exhibits superior stability, leading on more individual benchmarks than other methods. Given our high performance retention at a pruning cost only marginally higher than Wanda (as shown in "***Sec. 4.5. Pruning Efficiency Analysis***"), ATV-Pruning provides the optimal trade-off between performance and efficiency.

A4. Experimental Details for "*Sec. 3.2. Motivation Investigation: Modality-Aware Sensitivity Analysis*"

We detail the experimental setup for the motivation investigation in Sec. 3.2, involving the following aspects:

Adopted Benchmarks. Our sensitivity analysis is conducted on three standard LVLM benchmarks: MMBench-EN (MMB) [14], ScienceQA-IMG (SQA_{img}) [15], and VizWiz [8]. Consistent with our main experiments, all evaluations are managed via the `lmms-eval` toolkit [23] and adhere to the official metrics for each benchmark.

Implementation Details. All analyses use LLaVA-NeXT (8B) [13] as the backbone model. Activation statistics are estimated from a calibration pool of 128 high-quality image-text pairs from ShareGPT4V [3], identical to the calibration set used in our main experiments. The Mixture-of-Transformer (MoT) [5, 18] analysis probe is constructed by replicating the QKV and FFN linear layers within each

Transformer block to create distinct textual and visual pathways. A modality mask routes tokens to their respective pathway. Other components, such as residual connections, positional encodings, and the core attention logic, remain shared and identical. Pruning is then applied independently to each pathway under the three calibration configurations: text-only, image-only, and mixed.

A5. Implementation Details for "*Sec. 4.3.2 Discussion on Selection Strategy of Visual Tokens*"

Recalling Sec. 4.3.2, we evaluate two alternative saliency signals for visual token selection: *(i)* an **attention-based signal (ABS)** and *(ii)* a **diversity-based signal (DBS)**. These serve as ablative comparisons to our proposed visual drift metric (defined in Eq. 4). Here, we further provide their corresponding implementation details.

A5.1. Attention-based Signal (ABS)

For the ABS strategy, we replace the visual drift saliency score in Eq. 4. The new saliency s_v for a visual token v at block b is defined as the average cross-attention score it receives from all text tokens $\mathcal{T}$ in the prompt.

Let $A^{(b)}_{t\to v}$ be the attention weight from a text token $t \in \mathcal{T}$ to a visual token v in the cross-attention mechanism of block b. The saliency score is:

$$s_v = \frac{1}{|\mathcal{T}|} \sum_{t\in\mathcal{T}} A^{(b)}_{t\to v}$$

A higher s_v indicates that the visual token receives more attention from the text prompt, implying greater saliency. This attention-based s_v then replaces the visual drift score in all subsequent calculations: it is used to calculate the block-average saliency $\bar{s}$ (Eq. 5), determine the block-adaptive budget K (Eq. 6), and perform the final TopK selection (Eq. 7). This approach is inspired by prior attention-based visual token pruning work [4, 24].

A5.2. Diversity-based Signal (DBS)

The DBS strategy involves a two-part modification, replacing both the saliency definition (Eq. 4) and the selection mechanism (Eq. 7).

First, to maintain the block-adaptive budgeting framework, we define a per-token *diversity score* as the saliency s_v in Eq. 4. This score measures the average cosine distance of a visual token v from all other visual tokens $\mathcal{V}$ in the same sample, using their input representations $\mathbf{X}_{\text{in}}$:

$$s_v = \frac{1}{|\mathcal{V}|-1} \sum_{u\in\mathcal{V}, u\neq v} (1 - \cos(\mathbf{X}_{\text{in},v}, \mathbf{X}_{\text{in},u}))$$

This score s_v is used *only* to set the block-adaptive budget. Specifically, it is aggregated across all visual tokens to compute the block-average saliency $\bar{s}$ (Eq. 5), which in turn determines the per-sample budget K (Eq. 6).

Second, we replace the simple TopK selection (Eq. 7). Note that we do not simply take the TopK of these s_v scores. A high s_v only indicates high average diversity. It is possible for the K tokens with the highest s_v to be clustered together (e.g., all are distant from the majority of tokens, but form a small, tight cluster themselves). Therefore, to ensure the selected subset is *internally diverse*, we replace the TopK rule with a *max-min diversity selection scheme* [1]. This scheme directly selects the subset $\mathcal{V}_{\text{sub}}$ of size K that maximizes the minimum pairwise distance among its members, operating directly on the token representations. Formally, the selection rule (replacing Eq. 7) becomes:

$$\mathcal{V}_{\text{sub}} = \underset{\mathcal{S}\subset\mathcal{V}, |\mathcal{S}|=K}{\operatorname{argmax}} \left(\min_{u,w\in\mathcal{S}, u\neq w} d(\mathbf{X}_{\text{in},u}, \mathbf{X}_{\text{in},w}) \right)$$

where $d(\cdot,\cdot)$ is the cosine distance $1-\cos(\cdot,\cdot)$. This directly optimizes for a diverse subset, avoiding the issue where TopK selection might choose K tokens that are individually diverse on average but are all clustered in a similar region of the representation space, lacking internal diversity.

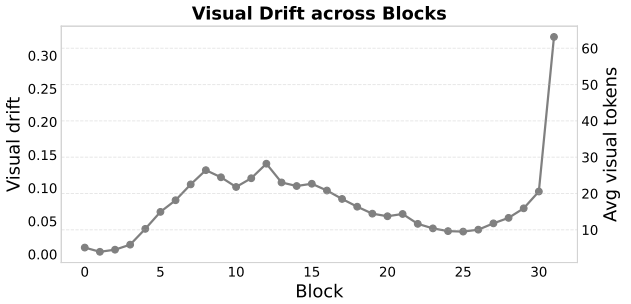

Figure A1. **Trend of ATV-Pruning's block-wise Visual Drift and Adaptive Token Allocation** on LLaVA-NeXT (8B) with $\alpha = 1.0$. The left y-axis measures the average visual drift per block. The right y-axis indicates the corresponding average number of visual tokens selected per sample. The trend illustrates the block-adaptive nature of our method: more tokens are retained in layers where visual representations undergo significant updates.

A6. An Extension of "*Sec. 4.3.2 Discussion on Selection Strategy of Visual Tokens*": Additional Visualization of Visual Drift across Blocks

To better understand the behavior of our proposed Block-Adaptive Visual Selection (Sec. 3.3.2), we further visualize the block-wise statistics of visual drift. Recall that we define visual saliency based on the representation drift, quantified as the cosine distance between the input and output features of a visual token within a block (Eq. 4). Consequently, the budget of visual tokens retained, K, is directly

Calibration	MMB	MME	SQA_{img}	VizWiz	Average
Visual-only	60.65	1591.69	66.93	60.73	90.4%
Mixed (default)	63.83	1620.02	69.06	60.73	92.6%
Text-only	64.69	1756.87	69.56	63.25	95.9%
ATV-Pruning	**65.29**	**1801.51**	**69.86**	**63.34**	**96.8%**

Table A3. Performance of pruning the *original* LLaVA-NeXT at 50% sparsity under different calibration pools. For Wanda, mixed calibration is the default setting, and text-only calibration gives the best retention among all vanilla calibration choices.

proportional to the average saliency $\bar{s}$ of that block (Eq. 5).

Fig. A1 illustrates the progression of visual drift across the 32 Transformer blocks of LLaVA-NeXT (8B) [13] with $\alpha = 1.0$ (default setting). We observe that visual drift is highly non-uniform across the blocks. Specifically, drift is relatively low in the early stages (Blocks 0–3), indicating stable feature representations. In contrast, the trend highlights two distinct zones of high activity: a broad elevation in the middle layers (peaking around Blocks 8–12) and a sharp, significant spike in the final layer (Block 31). By explicitly coupling the visual token budget K to this trend, ATV-Pruning acts dynamically: it aggressively minimizes visual tokens in "visually stable" blocks to allow essential text tokens to dominate the calibration pool, while automatically preserving a higher density of visual tokens for calibration in blocks where significant visual feature transformations occur.

A7. Further Justification for the MoT Probe

We emphasize that the MoT probe in Sec. 3.2 is used solely as an analysis tool to disentangle modality-specific sensitivity in LVLM pruning; all pruning results reported in Sec. 4 are obtained on the original LVLM. We further justify this design from the following three aspects.

(i) MoT does not distort LVLM behavior. Without pruning, the MoT probe is functionally equivalent to the original LVLM and produces mathematically identical outputs. Moreover, when only the visual pathway is pruned while keeping the textual pathway intact, the model retains over 99% performance (Tab. 1), suggesting that the probe itself does not introduce obvious artifacts.

(ii) The same calibration trend also appears on the original LVLM. To verify that our conclusions are not specific to the probe, we directly prune the original LLaVA-NeXT [13] with Wanda [20] under different calibration pools. As shown in Tab. A3, the default mixed calibration yields 92.6% average retention, while switching to text-only calibration improves it markedly to 95.9%; visual-only calibration performs the worst. Thus, the best-performing vanilla calibration choice for Wanda can be viewed as text-only, which further supports our main claim that text-anchored calibration is crucial for activation-aware pruning in LVLMs. ATV-Pruning still achieves the best overall performance, reaching 96.8% average retention and outperforming this oracle text-only calibrated Wanda. We also note that this behavior is specific to activation-aware Wanda-style pruning: in our experiments, we do not observe the same trend for SparseGPT [6], where mixed calibration remains the strongest choice on LVLMs.

(iii) The MoT probe provides additional diagnostic insight. Beyond recovering the same calibration trend, the MoT probe further explains why text-only calibration is already markedly better. In particular, it reveals that the textual pathway is the primary sensitivity bottleneck, making text calibration especially important for activation-aware pruning, while the visual pathway is considerably more redundant and can largely preserve performance even under text-only calibration. These observations motivate our final design choice in ATV-Pruning, i.e., anchoring calibration on all text tokens while supplementing them with a compact subset of salient visual tokens.

References

[1] 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. 3

[2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1

[3] 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. 2

[4] 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. 3

[5] 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. 2

[6] 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. 2, 4

[7] 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. 1

[8] 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. 1, 2

[9] 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. 1

[10] 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. 2

[11] 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. 1

[12] 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. 1

[13] 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

[14] 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. 1, 2

[15] 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. 1, 2

[16] 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. 1

[17] Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. Accelerating sparse deep neural networks. *arXiv preprint arXiv:2104.08378*, 2021. 1

[18] 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. 2

[19] 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. 1

[20] 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, 4

[21] 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. 1

[22] 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. 1

[23] 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. 2

[24] 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. 3