Libra-Merging: Importance-redundancy and Pruning-merging Trade-off for Acceleration Plug-in in Large Vision-Language Model

Longrong Yang[1]*, Dong Shen[2]*, Chaoxiang Cai[3], Kaibing Chen[2],
Fan Yang[2], Tingting Gao[2], Di Zhang[2], Xi Li[1]†
[1]College of Computer Science and Technology, Zhejiang University
[2]Kuaishou Technology
[3]School of Software Technology, Zhejiang University

In our supplementary material, we provide the following details and experiments:

- Sec. A: We provide experimental results on more datasets.
- Sec. B: We provide some additional discussion.
- Sec. C: We provide more implementation details.
- Sec. D: We provide more details about Video LVLM.

A. More Datasets

Some academic-task-oriented and instruction-following benchmarks are collected for evaluating the LVLM. For academic-task-oriented benchmarks, VQA-v2 [5] and GQA [7] assess the visual perception capabilities of models through open-ended short answers. VizWiz [6] evaluates the zero-shot generalization of models on visual questions asked by visually impaired people. ScienceQA [14], a multiple-choice benchmark, evaluates the zero-shot generalization of models on scientific question answering. TextVQA [15] focuses on text-rich visual question answering tasks.

For instruction-following benchmarks, POPE [10] evaluates the degree of hallucination in model responses on three sampled subsets of COCO [12]: Random, Common, and Adversarial. MME [2] assesses the visual perception of models with yes/no questions. MMBench [13] evaluates the robustness of model answers with all-round shuffling on multiple choice answers. MM-Vet [18] evaluates the model capabilities in engaging in visual conversations on a diverse range of tasks and evaluates the correctness and helpfulness of the responses using the GPT-4 evaluation framework.

As shown in Table A, we verify the effectiveness of Libra-Merging on 10 image-text benchmarks. For LLaVA-NeXT-8B, 47% Flops achieves a 0.2% average performance increase than 100% Flops by employing Libra-Merging.

B. Additional Discussion

B.1. Libra-Merging without Hierarchical Merging

Hierarchical Merging is not a core component of Libra-Merging. Thus, we remove it and conduct comparisons with existing methods to further demonstrate the superiority of Libra-Merging. As shown in Tab. B, experimental results consistently validate the effectiveness of Libra-Merging.

B.2. Extension to Qwen2-VL

To verify the generalizability of the findings, we extend Libra-Merging to Qwen2-VL. Qwen2-VL uses the naive dynamic resolution and multimodal rotary position embedding (M-RoPE), which is significantly different from the visual encoding of the LLaVA series models. As shown in the Tab. C, we report the experimental results on Qwen2-VL, which demonstrate the effectiveness of Libra-Merging across diverse model architectures.

*The authors contributed equally to this paper.
†Corresponding author.

Table A. LVLMs (image-text models) with different token compression methods on six benchmarks. We conduct experiments on three different LVLMs to verify the scalability of our method across different model sizes (7b *vs.* 13b) and visual token count (llava-1.5 *vs.* llava-next). Main evaluation Benchmarks include VQA^{v2} [4]; GQA [7]; VisWiz [6]; SQA^{I}: ScienceQA-IMG [14]; VQA^{T}: TextVQA [16]; POPE [9]; MME [2]; MMB: MMBench [13]; MM-Vet [17]. The Flops ratio 47% (37%) corresponds to compression ratio 50% (67%). T means trillion. We calculate the average performance across all datasets except for MME, naming it "Avg".

Model		Flops (T)	Ratio	VQA^{v2}	GQA	VisWiz	SQA^{I}	VQA^{T}	POPE	MME	MMB	MMB^{CN}	MM-Vet	Avg
LLaVA-1.5-7B	*vanilla*	3.82	100%	78.5	62.0	50.0	69.5	58.2	85.9	1512.0	64.7	58.2	31.1	**62.0**
	FastV	2.13	56%	77.7	60.4	50.8	68.8	57.6	83.2	1511.7	64.2	58.0	31.8	61.4
	Turbo	2.13	56%	77.8	61.6	50.7	68.7	57.4	85.8	1471.7	63.7	57.5	29.9	61.5
	Libra-Merging	1.78	47%	78.0	61.3	50.7	68.9	57.4	84.7	1502.5	64.3	58.5	31.5	61.7
LLaVA-1.5-13B	*vanilla*	7.44	100%	80.0	63.2	53.6	72.8	61.2	85.9	1531.3	68.5	63.6	36.1	**65.0**
	FastV	4.06	55%	79.5	62.7	54.2	73.0	60.8	85.4	1549.8	68.3	63.5	33.5	64.5
	Turbo	4.06	55%	79.5	62.8	54.4	72.7	60.7	86.1	1561.0	68.1	63.2	33.3	64.5
	Libra-Merging	3.47	47%	79.9	63.3	54.3	73.1	61.1	86.0	1531.1	68.4	63.7	33.8	64.8
LLaVA-NeXT-8B	*vanilla*	17.17	100%	82.8	65.9	52.5	77.3	69.8	86.2	1552.1	74.4	70.4	43.4	69.2
	FastV	9.36	55%	83.0	65.5	52.0	77.2	69.5	86.8	1572.6	74.5	70.6	43.1	69.1
	Turbo	9.36	55%	82.5	64.7	51.7	77.7	65.0	86.6	1505.3	73.4	69.1	43.0	68.2
	Libra-Merging	7.86	47%	83.0	65.7	52.4	77.6	69.7	86.9	1565.8	74.7	70.8	43.6	**69.4**

Table B. We remove the non-core component Hierarchical Merging and conduct fresh comparisons between Libra-Merging and existing methods. We prune 50% or 80% of visual tokens after Layer 3 during the compression process.

Model	Layer	R	GQA	SQA^{I}	MME	MMB	MMB^{CN}	TextVQA	Avg
LLaVA-1.5-7B	-	-	62.0	69.5	1512.0	64.7	58.2	58.2	62.5
+FastV	3	50%	60.4	68.8	1511.7	64.2	58.0	57.6	61.8
+Libra-Merging (no hier.)	3	50%	61.4	69.5	1513.1	64.3	58.6	57.3	62.2
+FastV	3	80%	56.6	69.0	1427.6	62.8	56.7	55.6	60.1
+Libra-Merging (no hier.)	3	80%	58.8	69.2	1440.0	62.5	57.4	55.6	60.7

Table C. Experimental results on Qwen2-VL. The visual encoding of Qwen2-VL significantly differ from that of LLaVA, so Qwen2-VL is suitable to be used for verifying the effectiveness of Libra-Merging across diverse model architectures.

Model	Layer	R	Nocaps	Flickr30k	GQA	POPE	MME	Avg
Qwen2-VL-7B	-	-	102.6	77.4	62.4	87.8	1683.6	82.5
+FastV	3	50%	102.8	76.7	60.5	86.8	1654.9	81.7
+Libra-Merging	3	50%	102.9	76.4	61.9	86.9	1690.3	81.9
+FastV	3	80%	98.8	69.0	55.0	81.8	1549.9	76.1
+Libra-Merging	3	80%	102.4	71.0	58.7	85.2	1650.1	79.3

B.3. Extension to Training

With flash-attention compatibility, we can extend token compression techniques to model training. Specifically, we conduct two novel technical improvements.

Firstly, we design a hybrid attention mechanism to replace flash-attention. Flash-attention is indispensable for accelerating attention computation, yet it does not output attention scores. Fortunately, our main goal is to preserve response-related visual information, which only requires attention scores between output token and visual tokens. Consequently, we compute only these attention scores, requiring merely $1 \times N_t$ score computations. Since FLOPs scale quadratically with token length, this introduces approximately $\frac{1}{N_t}$ additional FLOPs, which becomes negligible when $N_t \gg 1$.

The second challenge faced in compressing tokens during training is the training instability, leading to a noticeable per-

Table D. LVLMs (video-text models) with token compression on Video-MME during training. We compress 30% visual tokens at layer $\{4, 10, 16, 22\}$. "GPU Hours" means the total time needed for finishing the training.

Model		R	GPU Hours	Overall		Short		Medium		Long	
				w/o subs	w subs	w/o subs	w subs	w/o subs	w subs	w/o subs	w subs
VideoLLaMA-2 (7B)	*vanilla*	0%	391.2	49.8	54.7	58.0	63.6	47.0	53.1	44.3	47.3
	Libra-Merging	30%	168.1	50.1	54.4	58.4	63.4	48.4	52.7	43.4	47.2

Table E. **Study about hyper-parameter sensitivity.** Settings for results in Table A are highlighted in grey. The compression ratio R is set to 67% and we compress tokens at layer $\{7, 15, 23\}$.

(a) **The threshold for dividing tokens.**

τ	GQA	SQA^I	MME	MMB	MMB^{CN}	Avg
0.3	59.2	69.5	1491.6	63.6	58.1	62.6
0.5	60.2	69.4	1483.2	63.7	57.9	62.8
0.7	60.7	69.2	1480.1	63.9	58.2	63.0
0.9	59.1	69.3	1481.3	63.7	57.8	62.5

(b) **The temperature coefficient.**

η	GQA	SQA^I	MME	MMB	MMB^{CN}	Avg
1e-3	59.1	69.6	1493.1	63.7	57.7	62.5
1e-4	59.2	69.5	1492.4	63.7	57.8	62.6
1e-5	59.1	69.6	1489.0	63.7	57.9	62.6
$mean(\alpha)$	60.7	69.2	1480.1	63.9	58.2	63.0

formance drop. To solve this issue, we design a linear compression ratio reduction scheme at the end stage of training. Given the compression ratio R, we start reducing the compression ratio from 4500 and end at 5500, with the compression ratio being $\frac{5500-i}{1000} \cdot R$ for the iteration i.

We validate our training token compression scheme on the state-of-the-art VideoLLaMA-2 [1]. As shown in Table D, Libra-Merging maintains a competitive performance while reducing the GPU hours from 391.2 to 168.1.

B.4. Hyper-parameter Sensitivity

We conduct the sensitivity study of Libra-Merging in Table E for two main hyper-parameters, the threshold τ for dividing non-target tokens and the temperature coefficient η for obtaining merging weights. First, when the highest similarity between a token and target tokens $s^{max} > \tau$, it is grouped into the *positive set* to merge in target tokens directly; otherwise, it is grouped into the *negative set* to generate an information compensation token. We discuss different thresholds for dividing tokens: $\tau \in \{0.3, 0.5, 0.7, 0.9\}$. Second, we discuss different temperature coefficients $\eta \in \{$1e-3, 1e-4, 1e-5, $mean(\alpha)\}$, where $mean(\alpha)$ indicates that the temperature coefficient is the average of output attention of all tokens. As shown in Table E, we find: (i) Libra-Merging is relatively robust to different thresholds τ and temperature coefficients η. (ii) When τ = 0.7, the average performance is the best. When η = $mean(\alpha)$, the average performance is the best.

B.5. Ablation about Information Compensation Token

We position Information Compensation Token at the sequence end for implementation simplicity, and Information Compensation Token can be placed at any location theoretically. As shown in the Tab. F (v_c^{begin} for front-placed Information Compensation Token), Information Compensation Token works, while v_c^{begin} shows no gains. Then, since FLOPs scale quadratically with token length, adding one token to N_t tokens introduce approximately $\frac{2N_t+1}{N_t^2} \approx \frac{2}{N_t}$ additional FLOPs, which becomes negligible when $N_t \gg 1$.

B.6. Visualization

Visualization is helpful for understanding how the model compresses tokens. We follow Turbo to conduct the visualization. Specifically, we locate the position of each visual token in the image. If tokens are merged, these tokens are represented by the same color patch, while different tokens are represented by different color patches. As shown in Figure A, Turbo can merge response-related tokens and background tokens. Compared to Turbo, Libra-Merging keeps response-related tokens better (*e.g.*, "monitor" or "umbrella" tokens), thus having more accurate responses.

Table F. The ablation about Information Compensation Token. Information Compensation Token can bring a slight performance increase. Besides, the position of Information Compensation Token is not important.

Model	Layer	R	GQA	SQA^I	MME	MMB	MMB^{CN}	TextVQA	Avg
LLaVA-1.5-7B	-	-	62.0	69.5	1512.0	64.7	58.2	58.2	62.5
+Libra-Merging	3	50%	61.4	69.5	1513.1	64.3	58.6	57.3	62.2
+Libra-Merging *w/o* v_c	3	50%	61.3	69.3	1512.3	64.3	58.5	57.3	62.1
+Libra-Merging *w* v_c^{begin}	3	50%	61.3	69.5	1512.7	64.3	58.5	57.3	62.2

Figure A. Visualization samples about token merging. Libra-Merging keeps response-related tokens better (*e.g.*, "monitor" or "umbrella" tokens), thus having more accurate responses.

C. Implementation Details

FastV: First, we compute the output attention of each token as its importance metric. Then, we rank all visual tokens based on their importance metrics at layer K (K=3 in this paper). We keep the most important $R\%$ (R=50 in this paper) of visual tokens and discard the remaining visual tokens.

Turbo: First, we compute the output attention of each token as its importance metric. We also compute the redundancy r of each token as:

$$r_i = \text{Max}\{\mathcal{S}(v_i, v_j),\ \ j \in \{1, ..., N\} \backslash i\}, \tag{1}$$

where $\mathcal{S}(\cdot, \cdot)$ refers to cosine similarity, and Max is the maximum operation. The information degree of a token is $r - w \cdot \alpha$, where α is the output attention of each visual token. We follow Turbo [8] to set w=6.0. Then, we rank all visual tokens based on their information degree at layer K (K=3 in this paper). We name visual tokens having the highest $R\%$ (R=50 in this paper) information degree as non-target tokens and name the remaining visual tokens as target tokens. We averagely merge non-target tokens into their most similar target tokens. A little difference to Turbo [8] is that Turbo uses bipartite soft matching, which divides visual tokens into two partitions B and C of the same size; for each token in partition B, Turbo [8] keeps the highest cosine similarity concerning partition C as its redundancy; Turbo [8] sorts the information degree of B and merge the top $R\%$ tokens into C, by averaging merging the R% tokens in B into the corresponding tokens in C with the highest cosine similarity. The bipartite soft matching is faster but may fail to find the most similar token of a token, so we use greedy matching in this paper.

Libra-Merging: First, we compute the output attention of each token as its importance metric. Then, we split the visual token sequence into different intervals and select the most important tokens from each interval as target tokens. The remaining tokens serve as non-target tokens. We divide non-target tokens into *positive set* and *negative set* based on their similarities with target tokens. Then, we merge non-target tokens of positive set into target tokens. We condense non-target tokens of negative set into an information compensate token. All merging weights are generated from token importance; when a token is more important, it should have a higher merging weight.

D. Details about Video LVLM

VideoLLaMA-2 (7B) [11] is a state-of-the-art video LVLM model. Similar to LLaVA, it converts visual information into visual tokens and feeds these visual tokens into an LLM. The total layer number is 28 in VideoLLaMA-2 (7B). During

inference, we compress 75% visual tokens at layer 3 for FastV, Turbo, and Libra-Merging. During training, we compress 30% visual tokens at layers $\{4, 10, 16, 22\}$. We find that compressing too many visual tokens may lead to training instability. In each layer of VideoLLaMA-2 (7B), the attention layer consists of two linear layers with the size 3584×3584 and two linear layers with the size 3584×512. The FFN layer consists of a gate linear layer with the size 3584×18944, an up linear layer with the size 3584×18944, and a down linear layer with the size 18944×3584.

Video-MME [3] integrates video types from 30 fields, varied durations from 11 seconds to 1 hour, with multi-modal data and high-quality annotations. It includes 900 videos (254 hours) and 2,700 annotated question-answer pairs.

References

[1] 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. 3

[2] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 1, 2

[3] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*, 2024. 5

[4] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 2

[5] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 1

[6] 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

[7] 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, 2

[8] Chen Ju, Haicheng Wang, Haozhe Cheng, Xu Chen, Zhonghua Zhai, Weilin Huang, Jinsong Lan, Shuai Xiao, and Bo Zheng. Turbo: Informativity-driven acceleration plug-in for vision-language large models. In *European Conference on Computer Vision*, pages 436–455. Springer, 2025. 4

[9] 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. 2

[10] 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

[11] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 4

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1

[13] 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? *arXiv preprint arXiv:2307.06281*, 2023. 1, 2

[14] 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

[15] 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

[16] 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. 2

[17] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 2

[18] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 1