Conical Visual Concentration for Efficient Large Vision-Language Models

Supplementary Material

1. Comprehensive Comparison in Training

1.1. Baselines

To further validate the effectiveness of our method, we conduct comparisons using the identical training recipe as LLaVA-1.5-7B [4] with three other baselines: Q-Former [3], FastV [2], and LLaVolta [1]. These methods differ only in their compression strategies, while all other factors are kept consistent to ensure a fair comparison. These approaches represent the latest state-of-the-art techniques specifically designed for redundancy reduction. A brief introduction to each is provided below.

Q-former It is a lightweight transformer that uses a set of learnable query vectors to extract visual features from the frozen image encoder. This method has been used in many models and has proven to be effective. It is an efficient way to compress image tokens, with compression occurring before entering the LLM, thereby ensuring high compression efficiency.

FastV It is currently the state-of-the-art method for inference acceleration. Here, we also apply it for efficient training, as it leverages the image token redundancy in LLMs to accelerate inference, and there is no gap in directly transferring this approach to training.

LLaVolta This is the current state-of-the-art method for training acceleration. They introduce Visual Context Compressor, which reduces the number of visual tokens during training. To minimize information loss caused by the compression on visual tokens while maintaining training efficiency, they develop LLaVolta as a lite training scheme. LLaVolta incorporates stage-wise visual context compression to progressively compress the visual tokens from heavily to lightly, and finally no compression at the end of training, yielding no loss of information when testing.

1.2. Implementation details

Notably, the original implementation of FastV utilize an eager attention mechanism to facilitate the output of attention maps. However, this approach significantly slows down training. To ensure a fair comparison, we recalculate its attention maps using the FlashAttention implementation, as employed in our paper, to improve training speed. Additionally, since the original experiments for LLaVolta are conducted on 8× Nvidia RTX 6000 Ada GPUs, we retrain it on 8× NVIDIA A100 80GB GPUs to account for differences in training times across hardware. During inference, all image tokens were used for processing. For Q-Former, we increase the number of learnable query vectors from 32 to 288 to enhance its ability to understand images and reduce information loss during the image token compression process, ultimately achieving a training time comparable to other methods.

1.3. Main Results

From Table 1, it can be observed that our method incurs the lowest training and inference costs among all approaches. Specifically, the training time is only 76% of that required by vanilla LLaVA-1.5-7B, and the average tokens are reduced to just 46% of the original. LLaVolta and ViCo both achieve comparable performance on almost all benchmarks. However, LLaVolta achieves only limited reductions in training time, primarily due to its conservative approach to compressing image tokens, which gradually decreases the compression ratio during training. This strategy fails to effectively eliminate image information redundancy.

In comparison with FastV, our method demonstrates superior performance on nearly all benchmarks, including a 1.5% improvement on SQA and a 1.2% improvement on AI2D, while also achieving shorter training times, resulting in a double win. Furthermore, in inference, ViCo uses fewer FLOPs than FastV, indicating that our approach surpasses FastV in both training and inference stages. This indicates that FastV's early removal of image tokens inevitably leads to performance degradation, whereas ViCo's strategy is much more reasonable, retaining all critical image information in the shallow layers. Additionally, Q-former performs poorly across all benchmarks, primarily because the Q-former structure requires extensive pretraining data, which is insufficient in the LLaVA-1.5-7B training recipe to yield strong results.

2. Ablation Study about Stage S

In this section, we primarily discuss the ablation study of stages S. In these experiments, we set λ to 0.5, consistent with the previous experiments, and continue to follow the principle of evenly distributing layers within the LLM. If the entire LLM forward process is divided into more stages, the model will remove more image tokens at earlier layers, leaving fewer image tokens in the later layers of the LLM. Conversely, if fewer stages are used, the number of token compression steps during the forward process decreases, resulting in greater redundancy. This parameter is utilized to balance the performance and efficiency of ViCo.

2.1. Results Analysis

As shown in Table 2, we vary the number of stages from 3 to 5. Overall, the model's performance remains robust

Method	Average tokens	GPU hours	Infer Flops(T)	POPE	SQA	MMB	GQA	OCR VQA	SEED^I	MMStar	AI2D	Text VQA
LLaVA-1.5-7B	576	104 (100%)	3.82	85.9	66.8	64.3	62.0	59.8	66.1	33.2	55.6	58.2
Q-former	288	88 (84.6%)	1.89	67.2	66.9	53.8	41.3	19.0	49.2	28.6	51.8	44.4
FastV	306	81 (78.0%)	2.01	85.2	69.5	65.6	61.0	60.7	65.3	33.4	55.3	58.4
LLaVolta	339	93 (89.4%)	3.82	85.6	69.6	63.6	**62.2**	60.0	**66.3**	33.2	55.7	58.3
ViCo	**270**	**79 (76.0%)**	**1.78**	**86.0**	**71.0**	**66.1**	61.9	**61.0**	65.5	**34.0**	**56.5**	**58.5**

Table 1. **Compare ViCo with other efficient training strategies.** Average tokens here refer to the average image tokens across all LLM layers, while GPU hours represent the time required for model training. As shown in the table, our method achieves the best performance on nearly all benchmarks while also being the most cost-effective strategy in terms of both training and inference.

Model	λ	Stage	GPU hours	Infer Flops(T)	GQA	SEED^I	MMB	TextVQA	POPE	SQA
LLaVA-1.5-7B	vanilla	vanilla	104 (100%)	3.82	62.0	66.1	64.3	58.2	85.9	66.8
	0.5	3	85 (62.2%)	2.13	**62.0**	**66.1**	**66.2**	58.4	**86.2**	70.5
	0.5	4	79 (76.0%)	1.78	61.9	65.5	66.1	**58.5**	86.0	**71.0**
	0.5	5	75 (78.9%)	1.38	61.4	65.5	65.9	57.8	86.1	69.9

Table 2. **Ablation study results about stages S.** Dividing the LLM forward process into more stages causes the model to eliminate a larger number of image tokens in the earlier layers, leaving fewer tokens for processing in the later layers. On the other hand, using fewer stages reduces the number of token compression steps throughout the forward process, leading to increased redundancy. This parameter serves to balance the trade-off between the performance and efficiency of ViCo.

across these changes, demonstrating that our compression strategy is relatively well-designed and not overly sensitive to hyperparameters.

However, on more challenging benchmarks such as SEED Bench and TextVQA, a noticeable performance decline occurs when the number of stages is increased to 5. If stages are further increased, the model's performance clearly deteriorates. This is reasonable because, at the maximum stage setting of 32, ViCo would begin removing half of the image tokens right after the first layer, leaving only 2 image tokens by 8 layer, inevitably discarding critical image information.

Meanwhile, with stages set to 3 or 4, there is no significant performance drop. Therefore, we ultimately select $S = 4$, which strikes a balance between preserving performance and effectively pruning redundancy by concentrating the limited image tokens on the important regions of the image."

References

[1] Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. Llavolta: Efficient multi-modal models via stage-wise visual context compression. *arXiv preprint arXiv:2406.20092*, 2024. 1

[2] 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. *arXiv preprint arXiv:2403.06764*, 2024. 1

[3] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 1

[4] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 1