Appendices

In the appendices, we provide more detailed results in addition to our main paper, including additional ablation study, additional discussion on FlashAttention, and additional results on video benchmarks.

A. Addtional Ablation Study

Ablation study for additional frames. As shown in Table A, dense frames can improve the performance of VideoMME and MLVU. In comparison, our method not only reduces computation cost significantly, but also creates a token sequence with less redundancy, which in return further improves performance across all three benchmarks.

Ablation study for token merging across frames. We conducted additional experiments with temporal merging, where tokens from adjacent frames are merged iteratively. As shown in Table B, merging across frames leads to degraded performance, especially at lower retention ratios, validating our hypothesis in main paper that temporal merging disrupts the temporal order of tokens, resulting in a negative impact on performance.

Ablation study for base models. We conducted additional experiments with Qwen2-VL-7B-Instruct [69] and adjusted MAX_PIXELS to a smaller value due to GPU memory limitations. While LLaVA-Prumerge [62] is one of our baselines in the paper, it is not compatible with Qwen2-VL. This is because LLaVA-Prumerge assumes the use of an image-level CLS token, whereas Qwen2-VL encodes sampled video frames together and does not have an image-level CLS token. Results of FastV and our method applied to LLaVA-OV and Qwen2-VL are reported in Table C. Our method consistently outperforms the baseline while requiring fewer FLOPs and prefill time (same conclusion as Table 1 in paper). We also notice that the performance drop with Qwen2-VL is larger than with LLaVA-OV. It is likely due to the video encoder in Qwen2-VL, which mixes features of all frames. As shown in above paragraph, cross-frame token merging may disrupt temporal order and is less effective.

Model	VideoMME	MLVU	Egoschema
LLaVA-OV	58.2	64.7	60.1
LLaVA-OV + 128 frames	58.4	67.7	59.8
LLaVA-OV + 128 frames + Ours	**58.9**	**69.0**	**60.5**

Table A. Ablation study for dense frames on long video understanding.

Retention Ratio	50%	25%	12.5%	6.3%	3.1%
Temporal Merging	57.9	55.8	54.5	50.4	47.4
Spatial Merging (our default)	**58.5**	**58.0**	**56.6**	**53.6**	**52.3**

Table B. Ablation study for temporal or spatial merging on VideoMME.

Model	FLOPs	Prefill Time	VideoMME	MVBench	MLVU	Egoschema
LLaVA-OV	99.63	439.58	58.2	56.7	64.7	60.1
FastV [5]	21.24	79.56	55.9	55.9	61.1	57.5
Ours	**14.76**	**55.03**	**58.2**	**57.1**	**63.7**	**59.6**
Qwen2-VL	61.90	252.88	55.2	62.6	58.2	61.5
FastV [5]	14.07	51.11	51.2	57.7	54.2	57.7
Ours	**9.96**	**36.76**	**52.8**	**62.6**	**57.2**	**58.1**

Table C. Ablation study for different base models.

Inference	**FLOPs** (TB)	**Prefill Time** (ms)
Token Merging	22.90	83.93
Token Merging & FlashAttention	22.90	79.10
Token Merging & Token Pruning	14.76	55.03

Table D. Ablation of FlashAttention vs. pruning on VideoMME.

B. Additional Discussion on FlashAttention

Our method explores token merging and pruning for adaptive inference in multi-modal LLMs, a direction that is orthogonal to works on improving LLM efficiency, such as quantization [10], sparse attention [7], and efficient attention (*e.g.* FlashAttention [9]). Notably, our method is compatible with quantization and sparse attention, yet not with optimizations like FlashAttention (FA), where attention values are not explicitly computed. This is because, similar to prior work on token pruning [5], our method relies on attention values for selecting tokens. In Table D, we conduct a cost-benefit analysis to compare our token pruning with FA. Our method reduces much more prefill time than FA (e.g., -28.90 ms vs. -4.83 ms). In fact, even though FA improves the efficiency of attention mechanisms, with large token numbers, the computation cost remains high. Integrating the idea of FA and token pruning might be possible (*e.g.* sequence parallelism, matrix approximation), which we leave as future work.

Further, FA was introduced to reduce memory I/O access and accelerate computation, making it particularly beneficial for model training, where backward propagation demands substantial memory and compute resources. However, during inference, its advantages are less pronounced, and its use becomes optional. Instead, the number of tokens processed plays a more significant role in inference efficiency, as shown in Table D.

C. Additional Results on Video Benchmarks

Our method is characterized by the adaptive inference that can adjust accuracy-efficiency trade-offs based on contextual factors, such as the FLOP budget. Below, we present more results of adaptive inference on video benchmarks by

Model	FLOPs	Prefill Time	VideoMME	MVBench	MLVU	EgoSchema	NextQA	PerceptionTest
	(TB)	(ms)	wo / w-subs	test	m-avg	test	mc	val
Video LLMs								
LongVA-7B [92]	381.09	2186.04	52.6 / 54.3	-	56.3	-	68.3	-
LLaVA-OV-7B [33]	99.63	439.58	58.2 / 61.5	56.7	64.7	60.1	79.4	57.1
Training-free Method Applied during Inference								
LLaVA-Prumerge [62]	23.65	86.89	57.0 / 59.9	56.5	60.6	**61.0**	77.6	55.8
Ours	22.06	84.36	58.0 / **61.3**	**57.3**	**64.4**	59.8	78.3	**56.7**
Ours	**14.76**	**55.03**	**58.2 / 61.3**	57.1	63.7	59.6	**78.4**	56.0

Table E. Additional results on video benchmarks. Supported by our adaptive inference method, we can adjust the parameters in our method to achieve different accuracy-efficiency balance. In this table, we add one of our model variants that consumes more computation resources while achieving slightly better accuracy than our default model.

assuming a target FLOP budget.

In Table E, to match the computation cost of baseline method LLaVA-Prumerge (*i.e*., 23.65 FLOPs), we adjust the parameters of our method and create a model variant with comparable computation demand (*i.e*., 22.06 FLOPs). Despite fewer FLOPs, this model variant again largely outperforms LLaVA-Prumerge across most benchmarks (*e.g*., +1.0 on VideoMME, +3.8 on MLVU, +0.9 on PerceptionTest). Further, compared to our default model, this model variant achieves comparable performance on most benchmarks and slightly better results on others (*e.g*., +0.7 on MLVU, +0.7 on PerceptionTest). These results showcase the flexibility of our adaptive inference method, which can optimize the accuracy-efficiency trade-off to fit with specific contextual requirements.