An Empirical Study on How Video-LLMs Answer Video Questions

Supplementary Material

Our supplementary material is organized as follows: we first provide the experimental details, followed by additional experiments and results. Note that the numbering of figures and tables in this supplementary document continues from the main paper.

A. Experiment details

This section provides the complete experimental details for the main paper.

Implementation Details. We adopt the `lmms-eval` library [21] as a unified evaluation framework for all models, integrating our implementation into this engine as well as into the official repositories of the respective models to ensure consistency and fairness across experiments. All evaluations are conducted on a server equipped with 8 NVIDIA A100 GPUs. Specifically, we use the official `LongVA-7B` as the LongVA model, `InternVideo2_5_Chat_8B` as the InternVideo2.5 model, `llava-onevision-qwen2-7b-ov` as the LLaVA-OneVision model, `LLaVA-Video-7B-Qwen2` as the LLaVA-Video model, and `LLaVA-NeXT-Video-32B-Qwen` as the larger variant of LLaVA-Video. All models are evaluated using a consistent strategy, where 32 frames are uniformly sampled from each input video.

Task category of Video-MME: In fig. 4 of the main paper and Fig. 9 of the supplementary material, we plot the trends of four high-level tasks, which are reorganized from the original 12 tasks in Video-MME. The tasks are divided as follows: *Perception Task* includes Temporal Perception, Spatial Perception, and Attribute Perception. *Recognition Task* includes Action Recognition and Object Recognition. *Reasoning Task* includes Temporal Reasoning, Spatial Reasoning, Action Reasoning, and Object Reasoning. *Other Tasks* include Counting Problem, Information Synopsis, and OCR Problems. The performance of each high-level task is calculated as the average performance of all its subtasks.

Calculation of Attention FLOPs: For convenience, we ignore the system tokens. In the Potential Applications section of the main paper, we report the estimated attention FLOPs in Tab. 1. Here, we provide the detailed calculation process. We denote N as the number of video frames, M as the number of tokens per frame, P as the number of language tokens, and d as the hidden dimensions. The number of heads is omitted. For each attention layer, the overall attention matrix can be divided into four parts, with their FLOPs scaling approximately as follows:

$$\text{FLOPs(VS)} = 2 \times N \times M^2 \times d \tag{9}$$

$$\text{FLOPs(VT)} = 2 \times N \times (N-1) \times M^2 \times d \tag{10}$$

$$\text{FLOPs(LV)} = 2 \times 2 \times P \times N \times M \times d \tag{11}$$

$$\text{FLOPs(LL)} = 2 \times P^2 \times d \tag{12}$$

$$\begin{aligned}\text{FLOPs(total)} = \text{FLOPs(VS)} + \text{FLOPs(VT)} \\ + \text{FLOPs(LV)} + \text{FLOPs(LL)}\end{aligned} \tag{13}$$

It can be observed that the relative ratio of reduced computation in a particular component depends only on N, P, and M. For all experiments, we sample 32 frames, and thus $N = 32$ for all models. For different models, each image frame is processed into a different number of tokens, resulting in varying values of M: specifically, $M = 144$ for LongVA, $M = 21$ for InternVideo2.5 (comprising 16 vision tokens and 5 special tokens inserted before each frame, where every 5 special tokens are treated as one frame-equivalent token), $M = 196$ for LLaVA-OneVision, and $M = 169$ for LLaVA-Video. In typical VideoQA tasks, the number of language tokens P is relatively small—usually $P < 100$; for estimation purposes, we set $P = 100$. And the computation cost of video-related attention—particularly Video Temporal attention (VT)—dominates the total attention FLOPs. Take LongVA as an example: it has 28 layers. By exiting video tokens after layer 18 (i.e., retaining only Language-to-Language attention (LV) in later layers), the overall attention FLOPs can be reduced to 64.3% of the original. If temporal attention is further removed from the first 4 layers, the total computation cost drops to 47.8%.

B. Additional experiments results

Absolute Performance on Various Benchmarks. We report the performance ratio and layer ratio in fig. 3 and fig. 4 of the main paper for clearer visualization and systematic comparison across models with different performance levels and layer depths. Here we provide absolute performance values and the corresponding affected layer indices in Fig. 8 and Fig. 9 for LongVA, InternVideo2.5, LLaVA-OneVision, and LLaVA-Video-7B. These serve as complementary results to fig. 3 and fig. 4 of the main paper.

Additional Experiments on Application. We report the performance of LongVA and InternVideo2.5 under different settings in Tab. 2, serving as supplementary results to Tab. 1 in the main paper. As shown in Tab. 2, LongVA

Table 2. Performance of different models under various settings on three benchmarks. *Exit only* means that video tokens exit the model after a certain layer. *Exit + window* means that, in addition to exiting, we also control the temporal attention range—i.e., for certain layers, video frames are only allowed to perform spatial attention. For both LLaVA-OneVision and LLaVA-Video, we exit video tokens after layer 18 and limit the first 8 layers to perform spatial attention only, as they act as non-critical layers.

Model	Settings	Attention Flops	MME	MVBench	EgoSchema
LongVA	Baseline	100%	52.9	51.5	46.2
	Exit only	64.3%	52.8	50.8	46.2
	Exit + window	51.0%	53.0	51.5	46.6
InternVideo2.5	Baseline	100%	59.6	73.0	68.4
	Exit only	57.0%	59.4	72.4	68.6
	Exit + window	47.8%	58.6	72.1	67.6

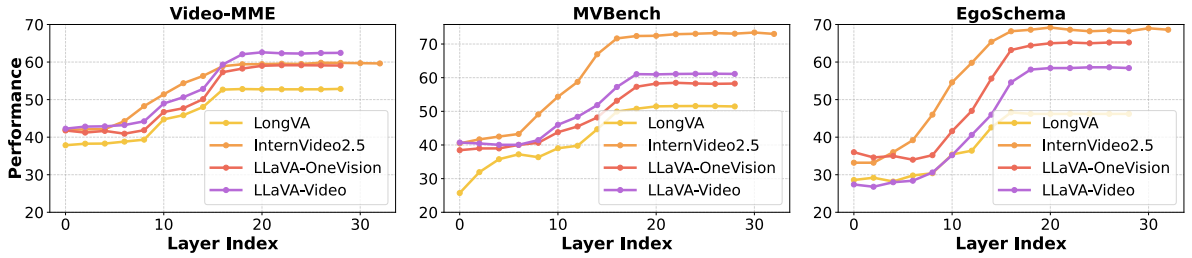

Figure 8. **Absolute Performance on different benchmarks with different models**. Setting: Setting: Apply Language-to-Video Knockout (LV-K) beyond a certain cutoff layer.

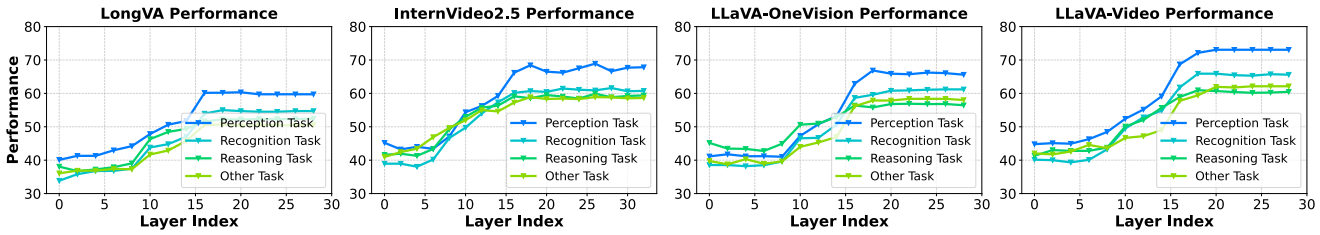

Figure 9. **Absolute Performance on different tasks of various models on Video-MME.** Setting: Apply Language-to-Video Knockout (LV-K) beyond a certain cutoff layer.

achieves the same performance while using only 51% of the original computation cost.

Additional Results on Finer-Grained Task-Level Categories. Here, we report the performance on the detailed 12 tasks. Overall, the performance of Video-LLMs follows a consistent trend: models without vision tokens achieve the best results, and our previous observation holds—exiting vision tokens after a certain layer often yields comparable, or even equivalent, performance to keeping all vision tokens. We present detailed results for all sub-tasks in VideoMME in Fig. 10, Fig. 11, Fig. 12, and Fig. 13, where most sub-tasks exhibit similar patterns. Notably, for certain models on specific subsets of tasks, early exiting of vision tokens even surpasses the full-token performance. For example, LongVA achieves better results on Temporal Perception, Temporal Reasoning, and Counting Problems (Fig. 10), while InternVideo2.5 outperforms its full-token baseline on Temporal Reasoning, Object Recognition, and Counting (Fig. 11).

C. Limitations

The scope of this work focuses on understanding the internal mechanisms and behaviors of how Video-LLMs answer video questions through empirical analysis. We acknowledge that this empirical study does not cover all existing leading Video-LLMs and VideoQA benchmarks. Also, this work centers on analysis, and we leave the development of more comprehensive solutions to future work. For instance, a promising direction is to develop dynamic, task-dependent inference strategies. Additionally, since our cur-

rent application does not involve any training, another valuable direction is to design and train Video-LLMs with more efficient architectures and attention mechanisms.

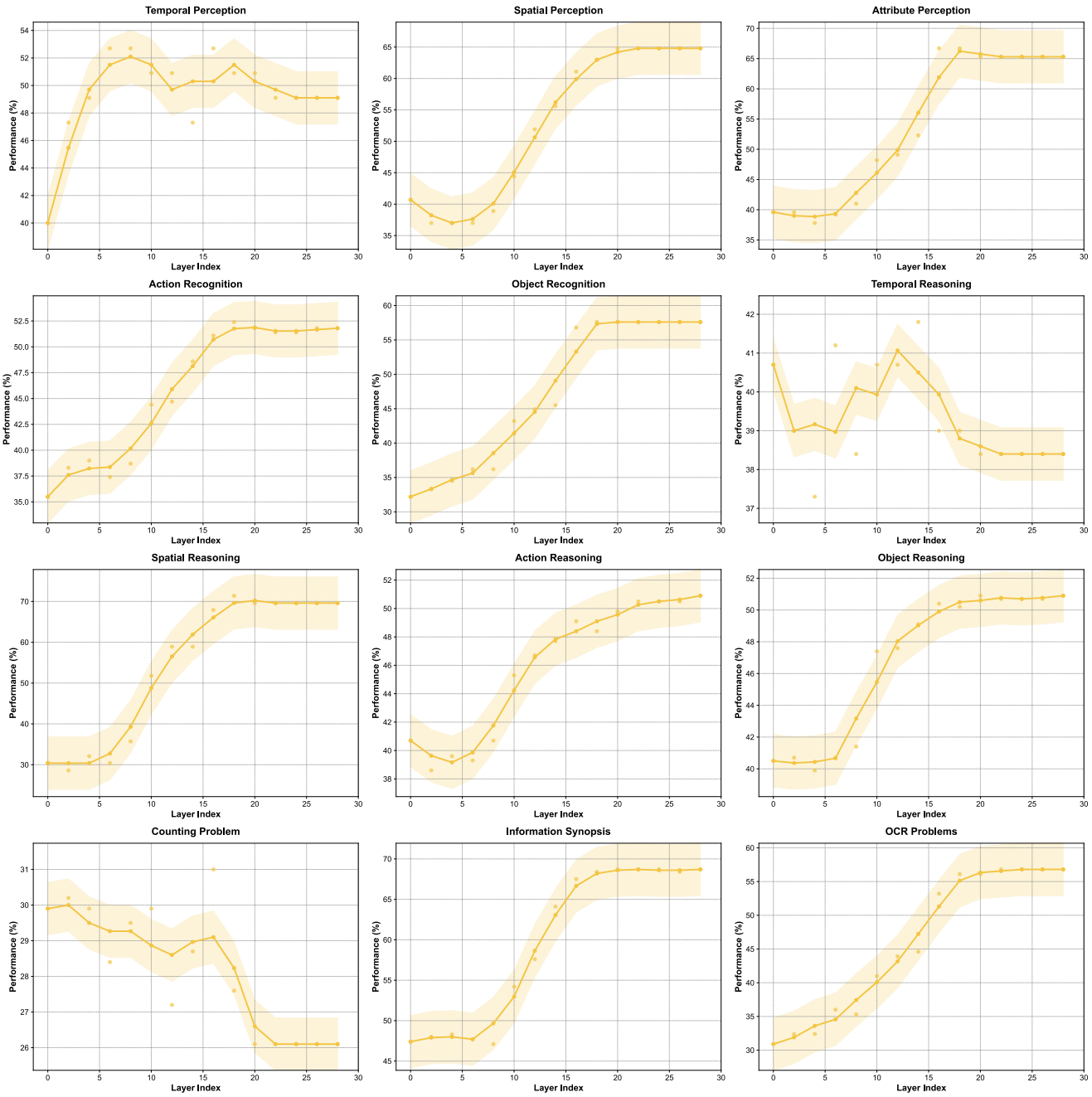

Figure 10. **Performance on different tasks of LongVA on Video-MME.** Setting: Setting: Blocking all video tokens after a certain layer ratio.

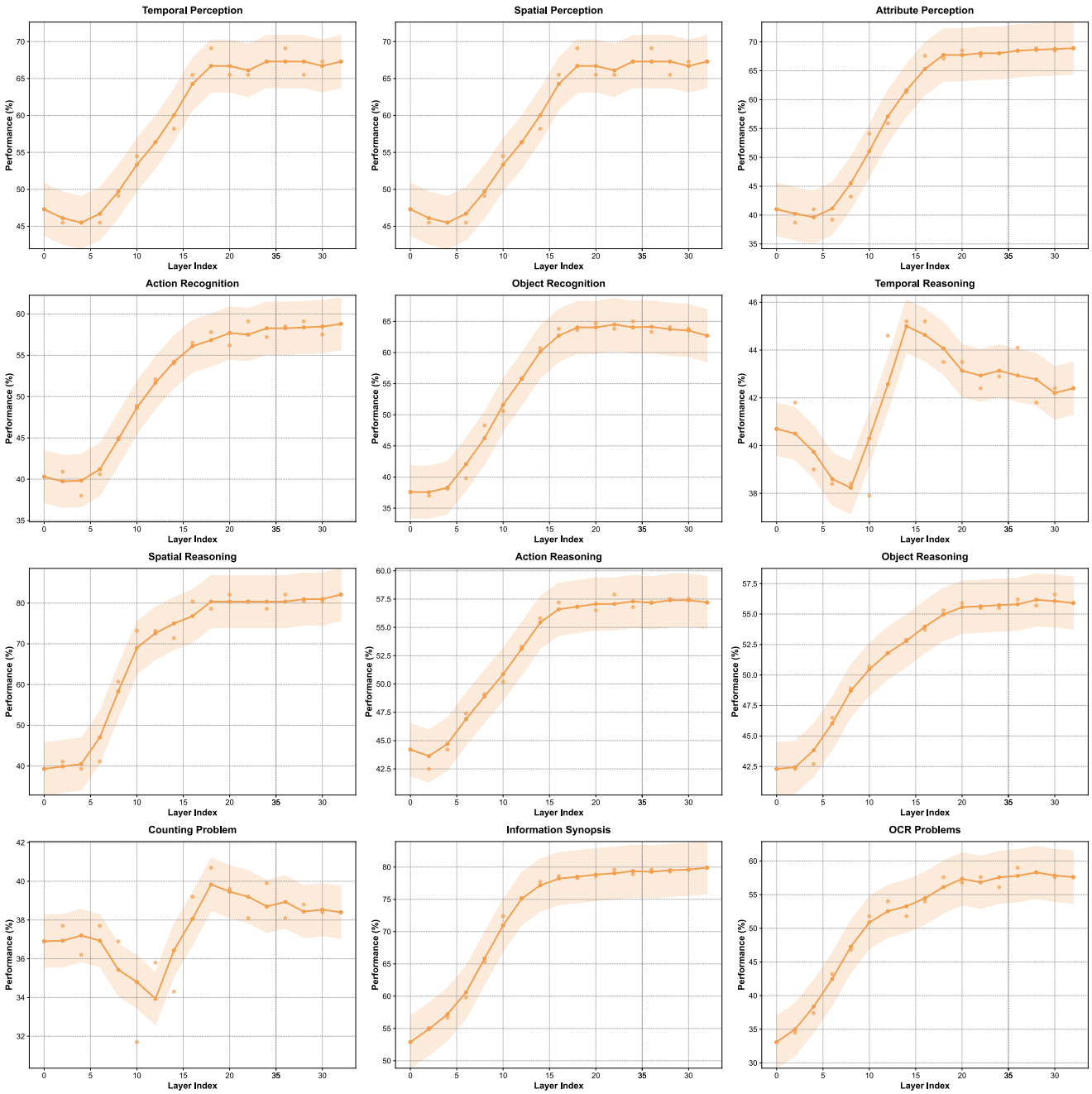

Figure 11. **Performance on different tasks of InternVideo2.5 on Video-MME.** Setting: Blocking all video tokens after a certain layer ratio.

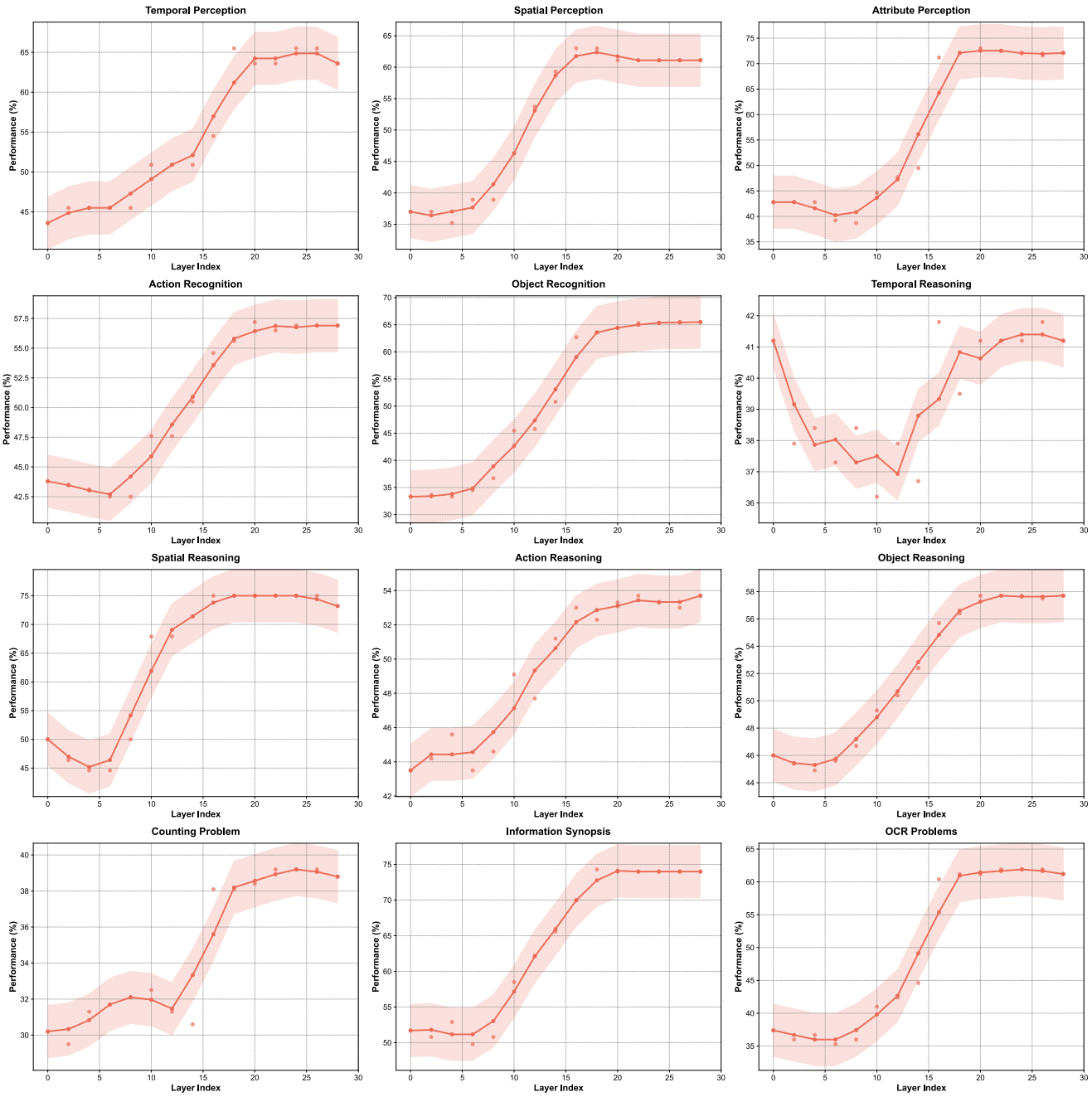

Figure 12. **Performance on different tasks of LLaVA-OneVision on Video-MME.** Setting: Blocking all video tokens after a certain layer ratio.

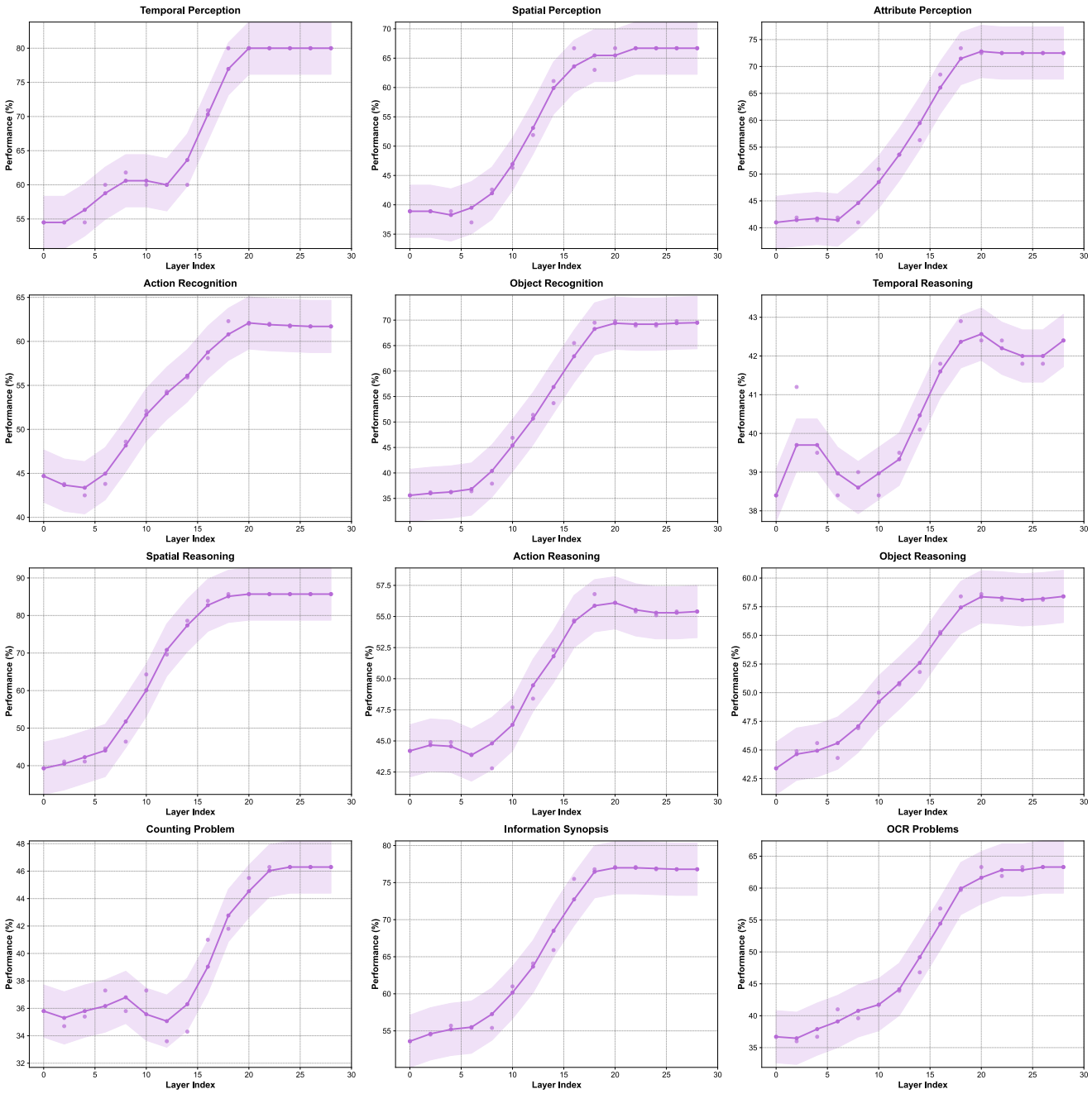

Figure 13. **Performance on different tasks of LLaVA-Video on Video-MME.** Setting: Blocking all video tokens after a certain layer ratio.