VideoLLM-MoD: Efficient Video-Language Streaming with Mixture-of-Depths Vision Computation

Shiwei Wu[1,2], Joya Chen[2], Kevin Qinghong Lin[2], Qimeng Wang[3], Yan Gao[3]
Qianli Xu[4], Tong Xu[1✉], Yao Hu[3], Enhong Chen[1✉], Mike Zheng Shou[2✉]

[1]University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
[2]Show Lab, National University of Singapore   [3]Xiaohongshu Inc.
[4]Institute for Infocomm Research, A*STAR

Abstract

A well-known dilemma in large vision-language models (*e.g.*, GPT-4, LLaVA) is that while increasing the number of vision tokens generally enhances visual understanding, it also significantly raises memory and computational costs, especially in long-term, dense video frame streaming scenarios. Although learnable approaches like Q-Former and Perceiver Resampler have been developed to reduce the vision token burden, they overlook the context causally modeled by LLMs (*i.e.*, key-value cache), potentially leading to missed visual cues when addressing user queries. In this paper, we introduce a novel approach to reduce vision compute by leveraging redundant vision tokens "skipping layers" rather than decreasing the number of vision tokens. Our method, VIDEOLLM-MOD, is inspired by mixture-of-depths LLMs and addresses the challenge of numerous vision tokens in long-term or streaming video. Specifically, for certain transformer layer, we learn to skip the computation for a high proportion (*e.g.*, 80%) of vision tokens, passing them directly to the next layer. This approach significantly enhances model efficiency, achieving approximately ~42% time and ~30% memory savings for the entire training. Moreover, our method reduces the computation in the context and avoid decreasing the vision tokens, thus preserving or even improving performance compared to the vanilla model. We conduct extensive experiments to demonstrate the effectiveness of VIDEOLLM-MOD, showing its state-of-the-art results on multiple benchmarks, including narration, forecasting, and summarization tasks in COIN, Ego4D, and Ego-Exo4D datasets. The code and checkpoints will be made available at github.com/showlab/VideoLLM-online.

1   Introduction

Recent advancements in large language models [65, 66, 6, 63, 72, 73, 56, 2], particularly with GPT-4o [62], have led many to believe that the development of a J.A.R.V.I.S.-like AI assistant is becoming increasingly feasible. Such an assistant would operate in a streaming manner, remain always-on, and be multimodal to facilitate interaction with users.

While existing video-based Large Multi-modal Models (LMMs) [54, 40, 84, 88, 69, 38, 57, 89, 81, 36, 20, 47] have shown significant capabilities in general visual content understanding and reasoning, these models primarily operate in an offline setting, provide response for a few sampled frames within a video in the event-level, which falls short in online settings where there is a need for prompt, concise, and frame-aligned answers for the continuous video frames, as shown in Figure 1. For

---

✉Corresponding Author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

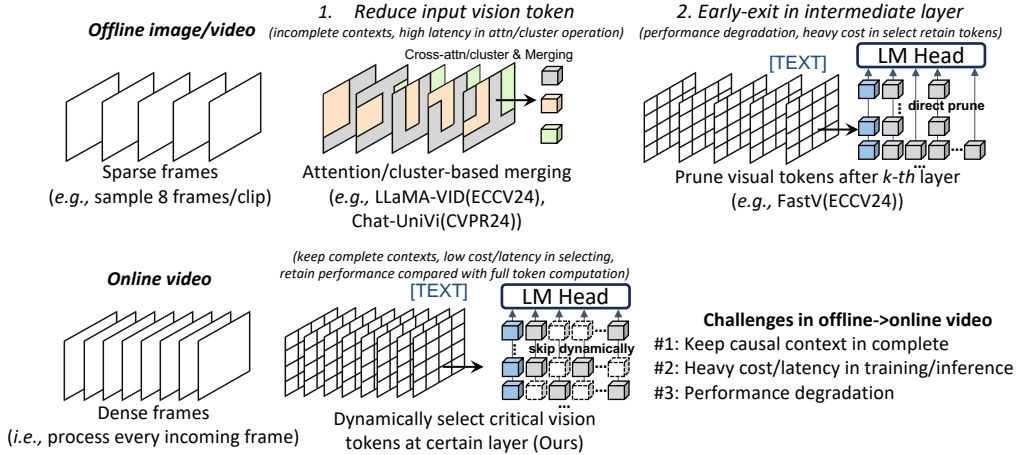



Figure 1: Comparison of efficient processing challenges in offline and online video. Online video introduces distinct challenges due to the need for real-time processing as frames stream continuously.

instance, in response to a query such as "remind me when I should add salt", the online assistant ought to evaluate each incoming frame and give temporal-aligned suggestions, taking into account historical visual and linguistic context, rather than merely summarizing the video at the event level. Consequently, as shown in Figure 1, online assistants face significant computational demands and challenges as they are required to engage in causal modeling for every frame of the long video, and current approaches [9] only rely exclusively on the CLS token for each frame, limiting the vision capability to spatial understanding, which is inadequate for scenarios that require fine-grained scene understanding.

It is intuitive to enhance spatial understanding by integrating additional pooled spatial tokens per frame. However, expanding vision resolution in the online scenario is challenging. Due to the dense attention mechanism and the deep layer design of existing LLMs, the training cost, including GPU memory and training time, increases quadratically as the number of vision tokens expands (*e.g.*, from 0.6k→6k vision tokens for a video consisting of 600 frames), which poses significant challenges to scaling up vision capabilities. Long videos, particularly online streaming videos, exhibit high redundancy in visual content, which suggests that a sparser approach could be used to process visual signals, potentially reducing the need for full attention in vanilla transformer-based LLMs without sacrificing performance.

Figure 2: Training Computation Cost. VIDEOLLM-MOD exhibits greater efficiency compared to the baseline.

In this paper, we propose VIDEOLLM-MOD, an efficient approach to scaling up vision resolution for online video large language models. Inspired by the Mixture-of-Experts (MoE) [21] and Mixture-of-Depth (MoD) [67] in LLMs, which utilizes conditional logic to route tokens to one of many expert feed-forward networks (FFNs) or certain intermediate layers, we propose that vision tokens in specific blocks can be either routed to subsequent self-attention and FFN operations or bypassed via residual connections. This approach, which we term *Mixture-of-Depth for vision tokens*, allows for the natural reduction of redundant vision information. Consequently, the model can learn which vision tokens are crucial, thereby optimizing computation accordingly. We surprisingly discovered that sparse operation at the token-level won't harm both vision capability and language modeling and is even better since it preserves original context as well as neglects the redundant vision signals, and it can dramatically reduce the training cost as shown in Figure 2, serving as "free lunch" in vision scaling. For each frame, instead of distributing FLOPs uniformly across all vision tokens in every decoder layer, we utilize a learnable module `LayerExpert` to allocate compute to critical vision tokens within the frame dynamically. Only a few vision tokens selected by the top-k route mechanism are processed by the following self-attention and FFN (Feed Forward Network) operations and the remains are skipped through residual connection. As shown in Figure 1, compared to directly dropping[10, 68]

2

vision tokens or merging [39, 20, 36] them to reduce computation, the *skip within context* mechanism preserves the completeness of the context, allowing for equivalent vision capability with significantly less computational effort. The proposed approach can also be generalized to traditional offline video settings seamlessly, such as COIN [71] and EgoExo4D [28] benchmarks.

We summarize our technical contributions as follows:

- We propose VIDEOLLM-MOD, an efficient approach to scaling vision resolution for online VideoLLM with reduced computational cost and similar or even better performance.
- We propose `LayerExpert` to determine which vision tokens should be processed at certain layers, leveraging the model to adaptively allocate computation to critical regions within incoming frames.
- Our experiments on Ego4D, EgoExo4D, and COIN benchmarks demonstrate the effectiveness and generalizability of our VIDEOLLM-MOD.

2 Related Work

Efficient Modeling in Transformer-based Models. The notorious squared complexity in vanilla Transformers [74] is a well-known problem, as it is one of the key bottlenecks in scaling the sequence length. In recent large multimodal models (LMMs) [1, 39, 50, 93, 63], prefix visual tokens are used as a fixed budget for context, significantly contributing to their efficiency. This issue becomes more pronounced in online video scenarios with denser video frames. Previous studies on large language models (LLMs)[79, 24, 10, 18] have explored the use of sparse computation to maintain performance during inference while reducing computational costs. However, these methods still incur significant training expenses. Efforts to reduce training costs through token pruning[68] and merging [36] techniques are not suitable for online scenarios, as they require offline computation to directly reduce token levels. Mixture-of-Depth [67] investigates the allocation of computation across model depth for language tokens, balancing performance with speed. Our model, VIDEOLLM-MOD, extends this approach to online video. We found that reducing vision computation in the context across model depth not only maintains but can even improve performance by removing high redundancy in video.

Large Multimodal Models for Online Video Understanding. Inspired by the success of numerous large language models (LLMs) [6, 63, 61, 85, 86], a series of large multimodal models (LMMs) [1, 39, 50, 93, 15] have subsequently been developed to further enhance our comprehension of the world. Current large multimodal models (LMMs) are capable of addressing a variety of standard benchmarks in video understanding, including temporal action localization [51], and video dialogue and question answering [40, 69, 54, 88, 44], while also demonstrating strong potential in broader multimodal applications [12, 13, 11, 46, 76, 77, 78]. However, while these models analyze entire video frames to make predictions in an "offline" setting, they are not optimized for real-time applications such as augmented reality (AR) glasses and autonomous driving systems. In light of this growing demand, benchmarks for online scenario such as action detection [90, 75] and anticipation [26, 89], which are designed to interpret events at the current timestamp without access to future data, are becoming increasingly critical. VideoLLM-online [9] serves as the first attempt to build an assistant using LLMs in an online video scenario. However, its spatial capabilities are limited, as it uses only a single CLS token to represent each frame, and expanding the vision scale is computationally expensive. VIDEOLLM-MOD proposes an efficient approach to scaling vision resolution by reducing vision computation in context, thereby enhancing spatial ability without incurring high computational costs.

Scaling up Vision Resolution for Large Multi-modal Models. Scaling up the visual resolution for LMMs is an effective approach to enhancing vision capabilities. By utilizing $5\times$ more vision tokens compared to LLaVA-1.5 [19], LLaVA-NeXT [49] achieved improved vision understanding. However, scaling vision tokens in online video scenarios presents significant challenges, as the training cost increases quadratically with the expansion of vision tokens, requiring the processing of every incoming frame in long videos. To handle long-context vision tokens in LMMs, CogAgent [31] integrates high-resolution image features into a low-resolution pathway via cross-attention across decoder layers. LLaMA-VID [20] utilizes context-attention to represent each frame with two key tokens. Both approaches are only applicable for offline video, as the high latency induced by the additional cross-attention mechanism is unacceptable in online scenarios. In contrast, VIDEOLLM-MOD receives streaming video-language input continuously and can reduce computation efficiently during every forward pass without additional overhead. This enables temporal-aligned responses, making it suitable for real-time applications.

3

3 Method

In this section, we introduce our VIDEOLLM-MOD framework, an efficient approach to training an online video large language model with a larger vision resolution.

3.1 Model architecture.

We depict the overall model architecture as shown in Figure 6, drawing parallels to LLaVA [50, 19, 49] in its design. The model is composed of three principal components: an image encoder, an MLP projector, and a language model. Each video frame embedding is represented as $(1 + h_p \times w_p) \times c$, which denotes the CLS token and the average pooled spatial tokens. The frame embeddings extracted by the image encoder are subsequently processed through the MLP projector to frame tokens. These tokens are interwoven with language tokens, forming the input for an LLM. We incorporate LoRA [32] in every linear layer of the LLM for efficient tuning. To select the most critical vision tokens, certain layers are also equipped with `LayerExpert` module, as detailed in Section 3.2.

Following VideoLLM-online [9], in addition to the language modeling (LM) loss, we also utilize an additional streaming loss to ensure the model remains silent when it is unnecessary to output responses. Both training objectives employ cross-entropy loss as follows:

$$L = \frac{1}{N} \sum_{j=1}^{N} (\underbrace{-l_{j+1} \log P_j^{[\texttt{Txt}_{j+1}]}}_{LM\,Loss} - \underbrace{\sigma s_j \log P_j^{[\texttt{EOS}]}}_{Streaming\,Loss}), \tag{1}$$

where l_j and s_j are condition indicators: l_j is 1 if the j-th token is a language response token, and 0 otherwise; s_j is 1 if both (1) the j-th token is the *last* token of a frame[2], and (2) $l_{j+1} = 0$. The streaming EOS loss is applied to frames prior to responding. $P_j^{[\texttt{Txt}_{j+1}]}$ represents the probability associated with the $(j + 1)$-th text token, as output by the language model head for the j-th token, while $P_j^{[\texttt{EOS}]}$ indicates that probability for the EOS token. The two objectives are balanced using the streaming loss weight σ.

3.2 Scale up Vision Resolution for Online Video.

Motivation. The performance of online assistants improves with increased vision scale (*i.e.,* $(1 + h_p \times w_p)$). However, enhancing vision resolution in online scenarios is challenging because the number of vision tokens grows with video duration, leading to quadratic computational complexity. As shown in Table 7, online videoLLMs must process every frame of long videos during both training and inference to maintain the integrity of the complete visual and linguistic historical contexts, which places significant demands on GPU memory and computational resources.

We hypothesize that videos exhibit high redundancy in temporal and spatial since consecutive frames often share a large portion of their content, especially if the frames are captured in quick succession. Just as humans continuously "see" their surroundings without always "focusing" on every visual detail, it is intuitive that we should skip some vision tokens in certain layers when processing them with a deep model.

Selecting vision tokens via `LayerExpert` in certain block. We leverage a per-block `LayerExpert` module to learn the selecting/routing behavior, *i.e.,* learn which vision tokens require more or less processing than others. The `LayerExpert` identify the "importance score" (in scalar weights) of each vision token within a frame, and only the top-k vision tokens are processed by the following operations. Notably, since the vision tokens of different frames in the online streaming scenario are processed in a causal manner, the top-k selection is performed at the frame level, meaning the top-k vision tokens are selected within each frame. The language tokens are always processed since they are much less redundant and significantly fewer in number compared to vision tokens.

Specifically, suppose we have a sequence of length N interleaved with n_t language tokens and n_v vision tokens. For the given layer l, the sequence $X^l = \{\text{Interleaved}(x_{t_i}^l, x_{v_i}^l) \mid 1 \leq t_i \leq n_t, 1 \leq v_i \leq n_v\}$. Within the $(1 + h_p \times w_p)$ vision tokens of each frame, the `LayerExpert` determines the

---

[2]Loss is applied only to the last token when a frame consists of multiple patch tokens.

4

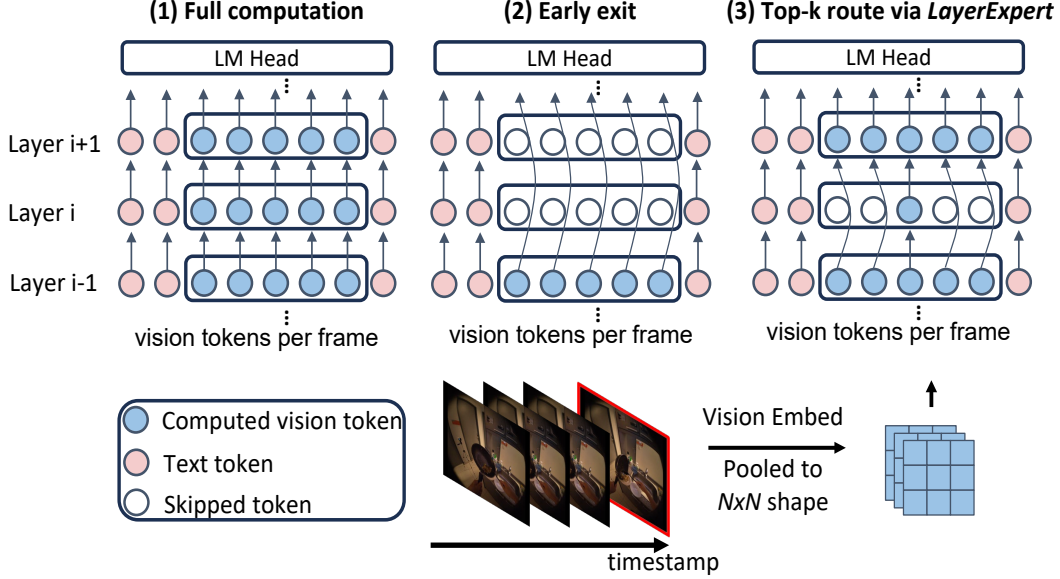



Figure 3: VIDEOLLM-MOD selects the top-k vision tokens within each frame in certain layers via `LayerExpert`. We observe that performance drops dramatically with *Early-exit* as critical vision tokens miss subsequent processing. By retaining crucial vision tokens in certain layers and reducing redundant tokens that may mislead understanding, VIDEOLLM-MOD achieves better performance with significantly lower computation costs compared to *Full-computation* baseline.

importance score μ for a given vision token using a linear projection $\mu_{t_i}^l = w_\theta^T x_{v_i}^l$. Then, vision tokens are selected based on a vision keep ratio r for following processing, and P_r^l is the $(1-r)$-th percentile among the weights μ of frame vision tokens. The block's output for the given vision token is as follows:

$$x_{v_i}^{l+1} = \begin{cases} \mu_{v_i} f_i(\hat{\mathbf{X}}^l) + x_{v_i}^l, & \text{if } \mu_{v_i} > P_r^l \\ x_{v_i}^l, & \text{if } \mu_{v_i} \leq P_r^l \end{cases} \quad (2)$$

where the $\hat{X}^l$ represents the interleaved tokens consisting of all language tokens and top-k vision tokens in layer l, and f_i denotes the subsequent self-attention and the FFN operations.

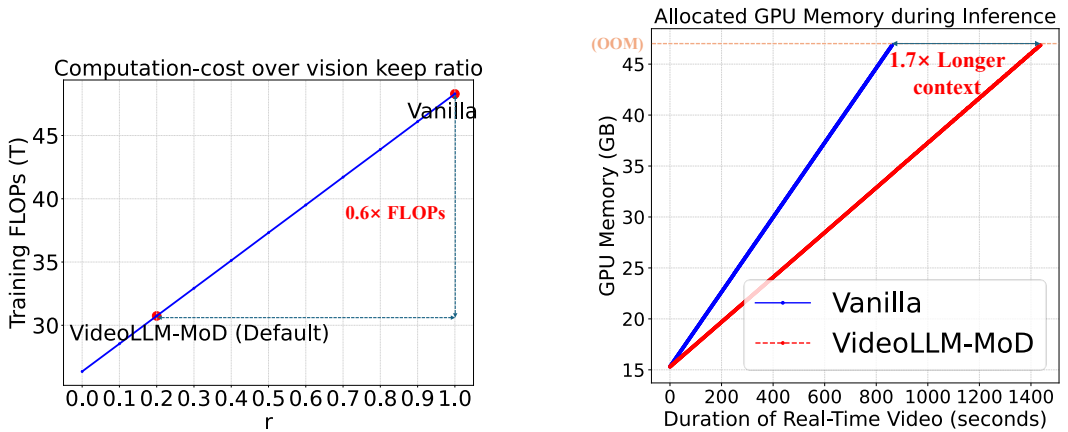
3.3 Efficiency analysis of VIDEOLLM-MOD.

We further analyze the computation cost of our approach. Except for the decoder layers, the other modules, including `LayerExpert`, the MLP projector, and LoRA, are fixed given certain inputs and are significantly smaller than the decoder layers of the language model. Therefore, we ignore their FLOPs computation and only consider the computation of the multi-head attention (MHA) and feed-forward network (FFN) modules in the FLOPs estimation.

Suppose the language model has L total hidden layers, in which d and m denote the hidden size dimension, and the intermediate size of FFN, respectively. The input sequence is interleaved with n_v vision tokens and n_t language tokens. We insert `LayerExpert` in K layers with a vision keep ratio r inside of the entire decoder layers. For each attention head, the theoretical FLOPs of the layer with `LayerExpert` is:

$$\text{FLOPs}_{\text{LayerExpert}} = 4(n_t + rn_v)d^2 + 2(n_t + rn_v)^2 d + 2(n_t + rn_v)dm, \quad (3)$$

while $r = 1$ in vanilla transformer layers. Since in the online video setting, vision tokens are significantly more than language tokens, *i.e.*, $n_v \gg n_t$, the FLOPs of entire decoder is proportional to the vision keep ratio r and the number of layers equipped with `LayerExpert` as follows:

5

(a) Calculated training FLOPs over vision keep ratio r. By default, VIDEOLLM-MOD only requires $0.6\times$ FLOPs compared to the *Full-computation* baseline.

(b) Allocated GPUMemory in inference phase. By saving the KV cache of historical states, VIDEOLLM-MOD supports $1.7\times$ longer video than baseline.

Figure 4: Efficiency analysis of VIDEOLLM-MOD in both training and inference phase.

$$\text{FLOPs}_{\text{decoder}} = \sum_{k=K+1}^{L} \text{FLOPs}_{\text{LayerExpert}}(r) + \sum_{k=K+1}^{L} \text{FLOPs}_{\text{vanilla}} \tag{4}$$

We further calculate the total FLOPs[3] of VIDEOLLM-MOD and the *Full-computation* baseline during the training phase in a real-world scenario. As shown in Figure 4a, the practical FLOPs of VIDEOLLM-MOD is only $0.6\times$ that of the baseline, and this value can be further reduced if the vision scale of each frame is larger, demonstrating the excellent scalability of our approach.

By skipping redundant vision tokens in certain layers, VIDEOLLM-MOD not only reduces training computation costs but also improves inference efficiency. As shown in Figure 4b, reducing the intermediate KVcache in historical states allows us to support $1.7\times$ longer context and achieve a comparative inference speed compared to baseline, facilitating deployment in real-world applications.

4 Experiments

4.1 Experimental Settings

Datasets. We validate the effectiveness of our proposed VIDEOLLM-MOD on both online and offline settings, including egocentric video dataset Ego4D [27] and EgoExo4D [28], as well as instructional video dataset COIN [71].

- **Ego4D Narration Stream Benchmark**: Following VideoLLM-online [9], we utilize the dense Ego4D timestamp-narrations to create a streaming set, aiming to generate timely narrations similar to those produced by Ego4D human annotators [27].

- **Ego4D long-term action anticipation (LTA) Benchmark**: This benchmark requires predicting the next $Z = 20$ actions (verbs and nouns) for a given video based on the previous 8 steps. Following previous studies [89, 33], we use the standard Ego4D v2 splits,

- **EgoExo4D Fine-grained Keystep Recognition Benchmark**: This task involves recognizing fine-grained key steps from procedural egocentric videos during the test phase, using models that can leverage multiple time-synchronized views during training.

- **COIN Benchmarks**: Following previous works [48, 91, 58, 83], we evaluate our model on six common benchmarks of the COIN dataset: step recognition, step forecasting, task summarization, procedure forecasting, and procedure forecasting with a goal.

---

[3]Calculated via DeepSpeed FLOPs Profiler, processing 600 frames with $(1+3\times3)$ patches on a single NVIDIA A100 GPU. The layers with `LayerExpert` are interleaved in every other layer in default.

6

Evaluation metrics and implementation details: For online benchmark, following VideoLLM-online [9], we use the Language Modeling Perplexity (*LM-PPL*) and *LM-Correctness* to evaluate the language modeling capability at the given timestamp. To evaluate the temporal alignment capability as an online assistant, we use the Time Difference (*TimeDiff*) and *Fluency* to comprehensively evaluate both the language modeling and temporal effectiveness. We trained all models on $8\times$ NVIDIA A100 GPUs. For each module, we use SigLIP-L/16 [87] as the visual encoder, a 2-layer MLP as the multimodal projector, and Meta-Llama-3-8B-Instruct [56] for the language model. For the vision embedding of each video frame, we use $(1 + 3 \times 3)$ tokens (CLS token + averaged pooled spatial tokens), with a frame rate of 2 FPS. We add LoRA [32] to all linear layers of the language model with a rank of 128 and a scaling factor of 256. Additional details can be found in Appendix A.3. For the trade-off between computation cost and performance, we insert `LayerExpert` every other layer and set the keep ratio r to 0.2 as the default setting.

4.2 Online Experiments

Method	Frame Strategy	FLOPs	Training Cost & Speedup	Ego4D Narration Stream on Validation Set			
				LM-PPL↓	*TimeDiff↓*	*Fluency↑*	*LM-Correctness↑*
VideoLLM-online [9]	1	5.75T	8hrs & n/a	2.43	2.04	45.1%	48.1%
Full-computation	1+3×3	48.29T	24hrs & n/a	2.40	2.05	45.3%	49.0%
EarlyExit		7.14T	10hrs & 2.4×	2.50	2.29	41.3%	46.2%
LayerSkip		26.35T	13hrs & 1.8×	2.52	2.24	42.0%	46.5%
VIDEOLLM-MOD		30.74T	14hrs & 1.7×	2.41	2.04	45.2%	48.9%

Table 1: Online experiments on the Ego4D narration benchmark. VIDEOLLM-MOD achieves comparable metrics to the *Full-computation* baseline with less computation cost.

We compare our VIDEOLLM-MOD model with various baselines on the Ego4D narration benchmark, as shown in Table 1. We analyze the baselines in detail as follows.

VideoLLM-online [9]. For a fair comparison, we re-implemented the VideoLLM-online [9] baseline using the same visual encoder and language model as VIDEOLLM-MOD in all experiments. By embedding each frame using only the CLS token, VideoLLM-online [9] achieves slightly worse performance on this benchmark due to the relatively simple narration, which does not heavily rely on fine-grained vision. Moreover, we found that larger vision resolution can indeed benefit performance, as shown in Figure 5, and in experiments that demand more detailed visual information as shown in Table 4, 5.

Full-computation. Using a vanilla transformer architecture, all vision tokens are processed densely across every layer, which significantly increases the training cost.

EarlyExit. Building on studies of language-only LLMs [25, 17], we adapt this approach to the online video setting. All vision tokens are processed in the shallow layers, then skipped in the deeper layers (equivalent to VIDEOLLM-MOD with $r = 1$ in the first few layers and $r = 0$ in the remaining layers). Empirically, we found that early exit at Layer 2 offers the best tradeoff between performance and computational cost, also highlighted in previous studies [79, 10]. This approach shows the lowest computation but the worst performance, as it misses most of the vision information.

LayerSkip. Introduced in previous LLM studies [18], we adapted the approach to the online scenario, skipping all vision tokens in every other layer (treated as VIDEOLLM-MOD interleaving layers with $r = 1$ and $r = 0$). Compared with VIDEOLLM-MOD, the performance drops significantly as critical vision tokens miss processing in certain layers.

Our **VIDEOLLM-MOD** exhibits the best tradeoff in online video scenarios, significantly reducing computational costs when processing excessive frames without sacrificing performance compared to the *Full-computation* baseline. Moreover, we discovered that our approach performs better than the *Full-computation* baseline in practical use, as shown in Figure 5. It seems counterintuitive that fine-tuning LLM with MoD performs better than using the vanilla model. The *dynamic layer skipping* methodology of the former results in less vision computation during the forward process, which is likely to weaken the spatial understanding capability. However, we argue that this increases the learning difficulty, as it forces the MoD gate at each layer to focus on the important vision tokens in current causal contexts. This may reduce the risk of overfitting and learn a more robust model.

7

4.3 Ablation Study

Insertion strategy of LayerExpert.

Table 2 presents the ablation study results for the insertion strategies of our LayerExpert.
We constructed different settings for inserting LayerExpert in the transformer layers. *All* and *Interleaved* refers to insert LayerExpert in every/every other layer. The *Interleaved* strategy demonstrates a better trade-off between computation cost and performance.

Insertion strategy	Fluency↑	LM-Correctness↑	FLOPs
All	42.5%	47.5%	13.18T
All-Deep	44.8%	48.8%	15.37T
Interleaved	45.2%	48.9%	30.74T
Interleaved-Deep	45.2%	49.0%	31.83T

Table 2: Ablations on the insertion strategy of LayerExpert in transformer layers. The *Interleaved* strategy strikes the best trade-off among the variants.

The postfix *-Deep* denotes that vision token skipping is performed only in the deep layers (*i.e.,* layers after Layer 2). Previous studies [79, 10] indicate that attention allocation across all tokens in the shallow layers (the first two layers) is much more balanced compared to the deep layers, making these shallow layers more vulnerable to token skipping. Our results with and without *-Deep* also indicate this phenomenon.

Selecting the critical vision token. As shown in Table 3, to validate the necessity and effectiveness of allocating computation to the crucial vision tokens, we created two variants that select vision tokens either randomly or uniformly. The poorer performance on *TimeDiff* indicates that the online capability is significantly impacted by missing critical vision information. This suggests that determining which vision tokens deserve processing is essential for maintaining performance while reducing redundancy.

We also conducted ablations on the number of vision tokens to retain based on the vision keep ratio r. Even with relatively fewer tokens and FLOPs, VIDEOLLM-MOD achieves satisfactory results, further demonstrating the critical vision selection capability of LayerExpert and highlighting the high redundancy present in the video.

Keep Strategy	Keep Ratio r	LM-PPL↓	TimeDiff↓	Fluency↑	LM-Correctness↑	FLOPs
Random	$r = 0.2$	2.45	2.18	43.6%	48.1%	30.74T
Uniform	$r = 0.2$	2.42	2.17	43.9%	48.6%	30.74T
Learnable	$r = 0.1$	2.43	2.11	44.7%	48.1%	28.54T
	$r = 0.2$	2.41	2.04	45.2%	48.9%	30.74T
	$r = 0.3$	2.41	2.05	44.9%	48.7%	32.93T

(Header: Ego4D Narration Stream Validation spans LM-PPL, TimeDiff, Fluency, LM-Correctness)

Table 3: Ablations on different vision selection strategies. Choosing which vision tokens to process is crucial for efficient vision computation allocation.

4.4 Offline Experiments

We demonstrate the generalizability of our proposed VIDEOLLM-MOD on traditional offline video scenarios, including recognition, summarization, and forecasting tasks. As shown in Table 5a, our method achieves the best performance compared to end-to-end baselines on the Ego4D LTA benchmark, with results only slightly lower than Palm [33] and AntGPT [89], which utilize EgoVLP [45] pretrained features followed by cascading performance enhancement methods.

Method	Not use HowTo100M	Step	Task	Next	Proc.	Proc.+
ClipBERT [37]	✓	30.8	65.4	-	-	-
TimeSformer [5]	✗	46.5	85.3	34.0	17.0	40.1
Paprika [92]	✗	51.0	85.8	43.2	-	-
DistantSup [48]	✗	54.1	90.0	39.4	-	41.3
VideoTF [58]	✗	56.5	91.0	42.4	40.2	46.4
ProcedureVRL [91]	✗	56.9	90.8	46.8	-	-
VideoTaskGraph [4]	✗	57.2	90.5	40.2	-	-
VideoLLM-online [9]	✓	62.5	92.2	49.3	48.6	53.3
Ours (Full-computation)	✓	63.1	92.7	49.1	49.8	**54.1**
Ours	✓	**63.4**	**92.8**	**49.7**	**49.8**	53.3

(Header: COIN Benchmark Top-1 Accuracy↑ spans Step, Task, Next, Proc., Proc.+)

Table 4: Results on COIN benchmarks (left to right): step recognition, task recognition, next forecasting, procedure forecasting, procedure forecasting with a goal.

By expanding the vision resolution, VIDEOLLM-MOD achieves state-of-the-art performance, significantly surpassing VideoLLM-online [9], which only uses the CLS token for each frame. This is particularly evident in tasks requiring complex spatial context understanding, such as the EgoExo4D Fine-grained Keystep recognition benchmark [28], as shown in Table 5b. Furthermore, our method achieves the best performance on most of the COIN benchmarks [71], as illustrated in Table 4, even outperforming our full-computation baseline. By adaptively focusing on processing critical vision

8

Method	Not use EgoVLP	End-to-end?	Ego4D LTA ED@Z=20↓ Verb	Noun	Action
CLIP [16]	✓	✓	0.739	0.769	0.941
EgoT2 [82]	✓	✓	0.722	0.764	0.935
I-CVAE [55]	✓	✓	0.753	0.749	0.931
HierVL [3]	✓	✓	0.724	0.735	0.928
VideoLLM [8]	✗	✓	0.721	0.725	0.921
Vi.LLM-o. [9]	✓	✓	0.691	0.692	0.889
Ours	✓	✓	**0.689**	**0.676**	**0.884**
Palm [33]	✗	✗	0.696	0.651	0.886
AntGPT [89]	✗	✗	0.650	0.650	0.877

(a) Results on Ego4D LTA benchmark, evaluated on public server. ED@Z=20 denotes editing distance for future 20 actions.

Method	Train data (v1)	Ego Accuracy (%) on Val (v1/v2)
TimeSFormer [5] (K600)	ego	35.25/35.13
TimeSFormer [5] (K600)	ego, exo	32.67/32.68
EgoVLPv2 [64] (Ego4D)	ego	36.89/36.51
EgoVLPv2 [64] (Ego4D)	ego, exo	37.03/35.84
EgoVLPv2 [64] (EgoExo)	ego	37.61/36.04
EgoVLPv2 [64] (EgoExo)	ego, exo	38.21/39.10
VI Encoder [59] (EgoExo)	ego, exo	40.23/40.34
Viewpoint Distillation [30]	ego, exo	37.79/38.19
Ego-Exo Transfer MAE [42]	ego, exo	36.71/37.17
VideoLLM-online [9]	ego	40.53/40.73
Ours	ego	**44.85/42.62**

(b) Results on EgoExo4D Fine-grained Keystep Recognition benchmark.

Table 5: Experiments on COIN [71] and Ego4D [27] benchmarks. VIDEOLLM-MOD achieves best results among end-to-end models.

tokens, VIDEOLLM-MOD not only reduces computation but also excels in spatial-temporal scene understanding, particularly in complex contexts.

4.5 Experiments on General Benchmarks

We further validate the effectiveness of our approach on general benchmarks as shown in Table 6,7, and 8. Our VIDEOLLM-MOD was trained using the same default settings, excluding streaming loss, and the same pretraining and fine-tuning data as LLaMA-VID [20].

Model	Frame strategy	Training Cost & Speedup	Overall	Short	Medium	Long
VideoLLaMA 2 [88]	32 frames & 32 tokens/frame	65hrs & **2.95×**	47.9	56.0	45.4	42.1
Chat-UniVi-v1.5 [36]	64 frames & 112 tokens/frame	53hrs & **2.41×**	40.6	45.7	40.3	35.8
Video-LLaVA [44]	8 frames & 49 tokens/frame	60hrs & **2.73×**	39.9	45.3	38.0	36.2
Ours	1fps & 10 tokens/frame	22hrs & n/a	**49.2**	**58.4**	**46.6**	**42.4**

Table 6: Results on Video-MME [23] benchmark. Despite requiring significantly less training time, VIDEOLLM-MOD still achieves top-tier performance.

Method	TFLOPs	Training Cost (Pretrain + Finetune)	GQA [34]	MME [22]	POPE [43]	SQA [53]
LLaMA-VID [20]	9.8	4hrs + 48.5hrs	64.3	1521.4	86.0	68.3
Ours	5.8	2hrs + 8.5hrs	62.8	1505.5	85.5	70.2

Table 7: Results on general image benchmarks.

Method	Training Cost (Pretrain + Finetune) & Speedup	MSVD-QA [7] Acc	Score	MSRVTT-QA [80] Acc	Score	ActivityNet-QA [29] Acc	Score
LLaMA-VID [20]	9hrs + 30hrs	69.7	3.7	57.7	3.2	47.4	3.3
Ours	8hrs + 14hrs & **0.56×**	**78.5**	**3.9**	**65.3**	**3.6**	**53.4**	**3.4**

Table 8: Results on general video benchmarks.

4.6 Visualization of VIDEOLLM-MOD.

We observe that VIDEOLLM-MOD performs more robustly compared to the VideoLLM-online [9] and *Full-computation* baselines. By selecting critical vision tokens, the learning difficulty increases, as this approach forces the MoD gate at each layer to focus on important vision tokens within current

9

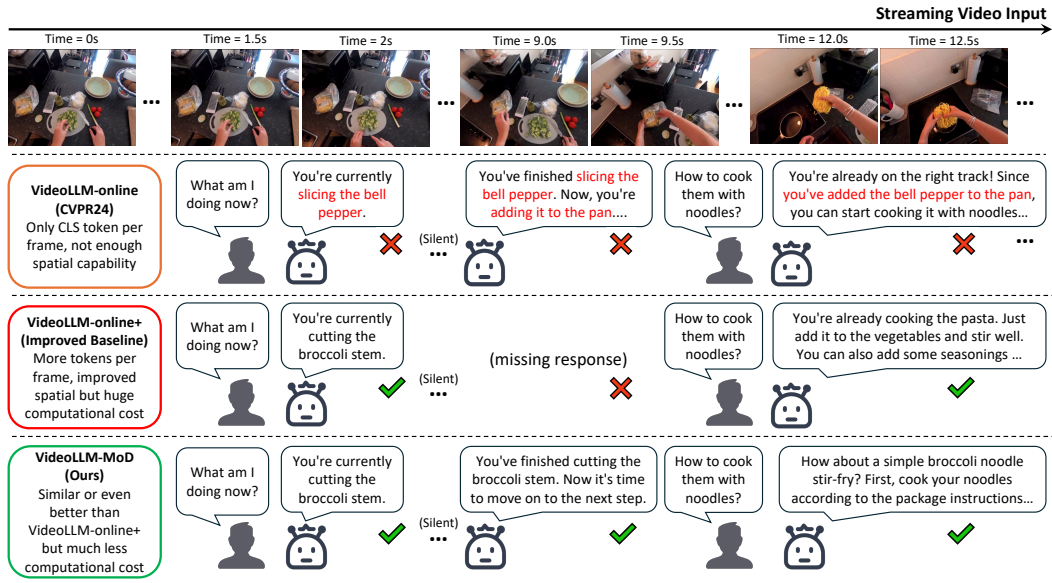
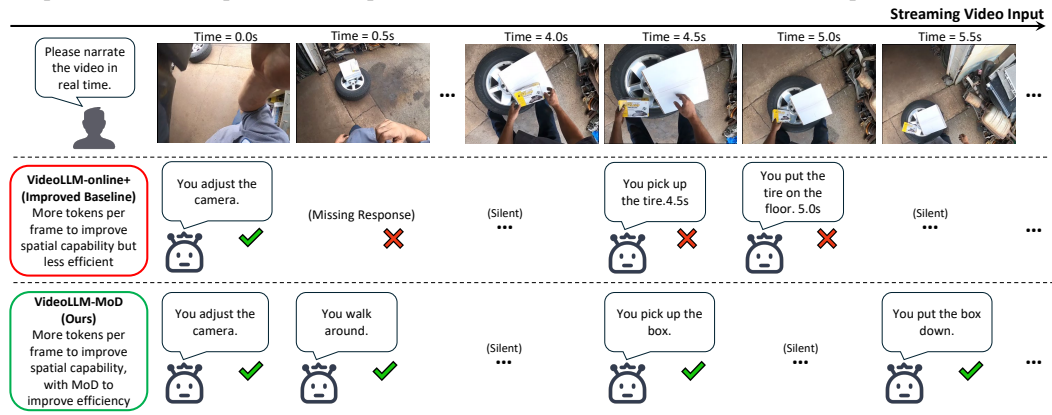
causal contexts. This strategy may reduce the risk of overfitting, thereby resulting in a more robust model as shown in Figure 5 cases.

(a) Using only CLS token often results in spatial understanding errors, *e.g.,* mistaking 'broccoli' for 'bell pepper.' VIDEOLLM-MOD improves fine-grained spatial ability by integrating more spatial tokens while reducing computation costs compared to the improved baseline. Text in red indicates incorrect response.

(b) We found that VIDEOLLM-MOD effectively reduces hallucinations and performs more robustly than the model trained with full computation. For instance, our model correctly recognizes "pick up the box" while the baseline mistakenly identifies it as "pick up tire."

Figure 5: Cases of VIDEOLLM-MOD on Ego4D GoalStep [70] video data.

5 Conclusion, Limitations, and Broader Impacts

In this paper, we introduced VIDEOLLM-MOD, which scales vision resolution for video large language models in online video through efficient select critical video tokens via `LayerExpert`. Our model can significantly reduce computational costs and memory usage with similar or even better performance compared with *Full-computation* baseline. Experiments on Ego4D, EgoExo4D, and COIN benchmarks confirm its efficacy and generalizability, making VIDEOLLM-MOD a robust solution for online video applications.

Limitations. As our primary focus was on developing an online assistant for ego-centric or instructional scenarios, we did not conduct extensive explorations on exo-centric video datasets.

Broader Impacts. Beyond the online scenario, we hope our work can provide insights and contribute to general video understanding tasks, particularly those involving long videos.

10

Acknowledgments and Disclosure of Funding

We thank Zhaoyang Lv from Reality Labs Research, Meta, as well as Ziteng Gao, Zechen Bai, and Zongbo Han from Show Lab, NUS, for their invaluable discussions on training VIDEOLLM-MOD. This work was supported in part by the grants from National Science and Technology Major Project (No. 2023ZD0121104), National Natural Science Foundation of China (No. 62222213, 62072423), and Program of China Scholarship Council (No. 202306340095). Mike Shou does not receive any funding for this work.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[2] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. Gemini: A family of highly capable multimodal models. *arXiv:2312.11805*, 2023.

[3] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. Hiervl: Learning hierarchical video-language embeddings. In *CVPR*, pages 23066–23078, 2023.

[4] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystep recognition in instructional videos. In *NeurIPS*, 2023.

[5] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021.

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, volume 33, pages 1877–1901, 2020.

[7] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011.

[8] Guo Chen, Yin-Dong Zheng, Jiahao Wang, Jilan Xu, Yifei Huang, Junting Pan, Yi Wang, Yali Wang, Yu Qiao, Tong Lu, et al. Videollm: Modeling video sequence with large language models. *arXiv preprint arXiv:2305.13292*, 2023.

[9] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *CVPR*, 2024.

[10] 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.

11

[11] Liyi Chen, Zhi Li, Weidong He, Gong Cheng, Tong Xu, Nicholas Jing Yuan, and Enhong Chen. Entity summarization via exploiting description complementarity and salience. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8297–8309, 2023.

[12] Liyi Chen, Zhi Li, Tong Xu, Han Wu, Zhefeng Wang, Nicholas Jing Yuan, and Enhong Chen. Multi-modal siamese network for entity alignment. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 118–126, 2022.

[13] Liyi Chen, Ying Sun, Shengzhe Zhang, Yuyang Ye, Wei Wu, and Hui Xiong. Tackling uncertain correspondences for multi-modal entity alignment. In *Proceedings of the 38th Conference on Neural Information Processing Systems*, 2024.

[14] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*, 2023.

[15] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C.H.Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv:2305.06500*, 2023.

[16] Srijan Das and Michael S. Ryoo. Video + clip baseline for ego4d long-term action anticipation. *arXiv:2207.00579*, 2022.

[17] Luciano Del Corro, Allie Del Giorno, Sahaj Agarwal, Bin Yu, Ahmed Awadallah, and Subhabrata Mukherjee. Skipdecode: Autoregressive skip decoding with batching and caching for efficient llm inference. *arXiv:2307.02628*, 2023.

[18] Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, et al. Layer skip: Enabling early exit inference and self-speculative decoding. *arXiv preprint arXiv:2404.16710*, 2024.

[19] Haotian Liu et al. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

[20] Yanwei Li et al. Llama-vid: An image is worth 2 tokens in llms. In *ECCV*, 2024.

[21] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, pages 1–39, 2022.

[22] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. MME: A comprehensive evaluation benchmark for multimodal large language models. 2023.

[23] Chaoyou Fu, Yuhan Dai, Yondong 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.

[24] Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive kv cache compression for llms. *arXiv:2310.01801*, 2023.

[25] Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *EMNLP*, pages 30–45, 2022.

[26] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. *Arxiv*, 2106.02036, 2021.

[27] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina González, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jáchym Kolár, Satwik Kottur,

Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbeláez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard A. Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3, 000 hours of egocentric video. In *CVPR*, pages 18973–18990, 2022.

[28] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, Eugene Byrne, Zachary Chavis, Joya Chen, Feng Cheng, Fu-Jen Chu, Sean Crane, Avijit Dasgupta, Jing Dong, María Escobar, Cristhian Forigua, Abrham Gebreselasie, Sanjay Haresh, Jing Huang, Md Mohaiminul Islam, Suyog Dutt Jain, Rawal Khirodkar, Devansh Kukreja, Kevin J. Liang, Jia-Wei Liu, Sagnik Majumder, Yongsen Mao, Miguel Martin, Effrosyni Mavroudi, Tushar Nagarajan, Francesco Ragusa, Santhosh Kumar Ramakrishnan, Luigi Seminara, Arjun Somayazulu, Yale Song, Shan Su, Zihui Xue, Edward Zhang, Jinxu Zhang, Angela Castillo, Changan Chen, Xinzhu Fu, Ryosuke Furuta, Cristina González, Prince Gupta, Jiabo Hu, Yifei Huang, Yiming Huang, Weslie Khoo, and et al. Ego-exo4d: Understanding skilled human activity from first- and third-person perspectives. *arXiv:2311.18259*, 2023.

[29] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, pages 961–970, 2015.

[30] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[31] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*, 2023.

[32] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[33] Daoji Huang, Otmar Hilliges, Luc Van Gool, and Xi Wang. Palm: Predicting actions through language models @ ego4d long-term action anticipation challenge 2023. *arXiv:2306.16545*, 2023.

[34] 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.

[35] HuggingFaceM4. Introducing idefics2: A powerful 8b vision-language model for the community. https://huggingface.co/blog/idefics2, 2024.

[36] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. *arXiv preprint arXiv:2311.08046*, 2023.

[37] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, pages 7331–7341, June 2021.

[38] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A multi-modal model with in-context instruction tuning. *arXiv:2305.03726*, 2023.

[39] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

13

[40] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv:2305.06355*, 2023.

[41] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. *arXiv preprint arXiv:2311.17005*, 2023.

[42] Yanghao Li, Tushar Nagarajan, Bo Xiong, and Kristen Grauman. Ego-exo: Transferring visual representations from third-person to first-person videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6943–6953, 2021.

[43] 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.

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

[45] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhong-cong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, Chengfei Cai, Hongfa Wang, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pretraining. *arXiv:2206.01670*, 2022.

[46] Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jin-peng Wang, Rui Yan, and Mike Zheng Shou. Univtg: Towards unified video-language temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2794–2804, 2023.

[47] Kevin Qinghong Lin, Pengchuan Zhang, Difei Gao, Xide Xia, Joya Chen, Ziteng Gao, Jinheng Xie, Xuhong Xiao, and Mike Zheng Shou. Learning video context as interleaved multimodal sequences. *arXiv preprint arXiv:2407.21757*, 2024.

[48] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *CVPR*, pages 13843–13853, 2022.

[49] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[50] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 2023.

[51] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. *arXiv preprint arXiv:2404.00308*, 2024.

[52] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

[53] 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.

[54] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv:2306.05424*, 2023.

[55] Esteve Valls Mascaro, Hyemin Ahn, and Dongheui Lee. Intention-conditioned long-term human egocentric action anticipation. In *WACV*, pages 6037–6046, 2023.

[56] Meta. Build the future of ai with meta llama 3. https://llama.meta.com/llama3/, 2024.

[57] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *arXiv:2305.15021*, 2023.

14

[58] Medhini Narasimhan, Licheng Yu, Sean Bell, Ning Zhang, and Trevor Darrell. Learning and verification of task structure in instructional videos. *arXiv:2303.13519*, 2023.

[59] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[60] OpenAI. GPT-4 technical report. *arXiv:2303.08774*, 2023.

[61] OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt/, 2023.

[62] OpenAI. Hello gpt-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[63] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

[64] Shraman Pramanick, Yale Song, Sayan Nag, Kevin Qinghong Lin, Hardik Shah, Mike Zheng Shou, Rama Chellappa, and Pengchuan Zhang. Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. In *ICCV*, 2023.

[65] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[66] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[67] David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

[68] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024.

[69] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Xun Guo, Tian Ye, Yan Lu, Jenq-Neng Hwang, et al. Moviechat: From dense token to sparse memory for long video understanding. *arXiv:2307.16449*, 2023.

[70] Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. In *NeurIPS*, 2023.

[71] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, pages 1207–1216, 2019.

[72] 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:2302.13971*, 2023.

[73] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*, 2023.

15

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 6000–6010, 2017.

[75] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. Oadtr: Online action detection with transformers. In *ICCV*, pages 7545–7555, 2021.

[76] Shiwei Wu, Joya Chen, Tong Xu, Liyi Chen, Lingfei Wu, Yao Hu, and Enhong Chen. Linking the characters: Video-oriented social graph generation via hierarchical-cumulative gcn. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4716–4724, 2021.

[77] Shiwei Wu, Weidong He, Tong Xu, Hao Wang, and Enhong Chen. Winning the cvpr'2022 aqtc challenge: A two-stage function-centric approach. *arXiv preprint arXiv:2206.09597*, 2022.

[78] Shiwei Wu, Chao Zhang, Joya Chen, Tong Xu, Likang Wu, Yao Hu, and Enhong Chen. From a social cognitive perspective: Context-aware visual social relationship recognition. *arXiv preprint arXiv:2406.08358*, 2024.

[79] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv:2309.17453*, 2023.

[80] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016.

[81] Lin Xu, Yilin Zhao, Daquan Zhou, Zhijie Lin, See Kiong Ng, and Jiashi Feng. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*, 2024.

[82] Zihui Xue, Yale Song, Kristen Grauman, and Lorenzo Torresani. Egocentric video task translation. In *CVPR*, pages 2310–2320, 2023.

[83] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. Unloc: A unified framework for video localization tasks. In *ICCV*, pages 13623–13633, 2023.

[84] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*, pages 10714–10726, 2023.

[85] Mingjia Yin, Hao Wang, Wei Guo, Yong Liu, Suojuan Zhang, Sirui Zhao, Defu Lian, and Enhong Chen. Dataset regeneration for sequential recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3954–3965, 2024.

[86] Mingjia Yin, Chuhan Wu, Yufei Wang, Hao Wang, Wei Guo, Yasheng Wang, Yong Liu, Ruiming Tang, Defu Lian, and Enhong Chen. Entropy law: The story behind data compression and llm performance. *arXiv preprint arXiv:2407.06645*, 2024.

[87] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, pages 11941–11952. IEEE, 2023.

[88] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv:2306.02858*, 2023.

[89] Qi Zhao, Ce Zhang, Shijie Wang, Changcheng Fu, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Antgpt: Can large language models help long-term action anticipation from videos? In *ICLR*, 2024.

[90] Yue Zhao and Philipp Krähenbühl. Real-time online video detection with temporal smoothing transformers. In *ECCV*, pages 485–502, 2022.

[91] Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. Learning procedure-aware video representation from instructional videos and their narrations. In *CVPR*, pages 14825–14835, 2023.

16

[92] Honglu Zhou, Roberto Martín-Martín, Mubbasir Kapadia, Silvio Savarese, and Juan Carlos Niebles. Procedure-aware pretraining for instructional video understanding. In *CVPR*, pages 10727–10738, 2023.

[93] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv:2304.10592*, 2023.

17

A Appendix

A.1 Online Video Demo.

We made several demo videos to showcase VIDEOLLM-MOD's effectiveness, available on the anonymous website https://sites.google.com/view/videollm-mod-anonymous.

A.2 Model Architecture

We depict the overall model architecture as shown in Figure 6, which is composed of three principal components: an image encoder, an MLP projector, and a language model. Following VideoLLM-online, we train the model with both language model loss and streaming loss.

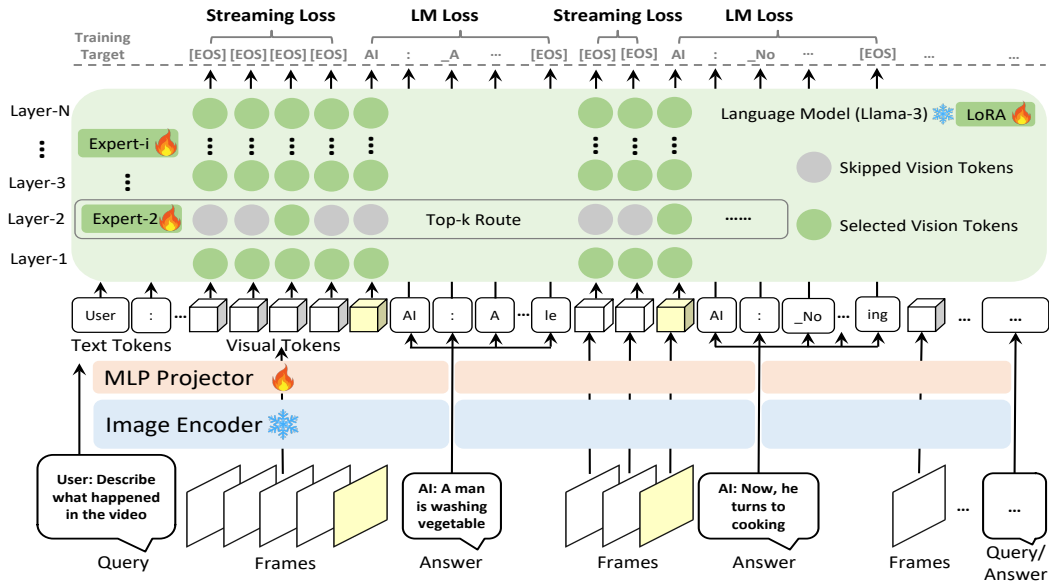



Figure 6: Model architecture of VIDEOLLM-MOD, ⬡ ⬡ ⊤ represent vision, keyframe, and text tokens, respectively.

A.3 Implementation Details

For a fair comparison, we trained the models on the Ego4D narration benchmark for 2 epochs with a learning rate of 2×10^{-4}. For the Ego4D LTA benchmark, EgoExo4D fine-grained keystep recognition benchmark, and Coin benchmark, we trained the models for 6, 10, and 5 epochs with learning rates of 3×10^{-4}, 2×10^{-4}, and 1×10^{-4}, respectively. During training, we set the batch size to 64 and streaming loss weight σ to 1.0 by default. For the trade-off between computation cost and performance, we insert `LayerExpert` every other layer and set the keep ratio r to 0.2 as the default setting.

A.4 Vision Strategy in Popular Large Multimodal Models.

We summarize the vision strategy employed in popular large multimodal models (LMMs) as shown in Table 7. The number of vision tokens increases continuously with video duration, as online video LMMs demand processing for every frame of the entire video. This leads to a quadratic increase in computational requirements during training and inference.

A.5 Online correction via data augmentation on temporal.

During streaming inference, we find that the model may be overfitted to the previous text output, likely due to insufficient training data in the LiveChat [9] dataset. To address this problem, we

18

Different LMMs		TemporalSample	#Token/frame→#Total VTokens
Image-based (single image)	BLIP-2 [39]	-	$32 \rightarrow 32$
	LLaVA [50]	-	$256 \rightarrow 256$
	LLaVA-1.5 [19]	-	$576 \rightarrow 576$
	LLaVA-NEXT [49]	-	$5 \times 576 \rightarrow 2880$
	DeepSeek-VL [52]	-	$576 \rightarrow 576$
	Idefics2 [35]	-	$5 \times 64 \rightarrow 320$
	GPT-4V [60]	-	765 (for 1024^2 Res.)
Offline video-based ($5 \sim 35s$)	Video-LLaVA [44]	8	$256 \rightarrow 2048$
	VideoChat2 [41]	8	$96 \rightarrow 768$
	Chat-UniVi [36]		merge to 112 S-T tokens
	Video-ChatGPT [54]		merge to 356 S-T tokens
Online/long video-based ($\sim 300s$, e.g., 2fps)	LLaMA-VID [20]	Use all	$2 \rightarrow 1.2k$ (LongLoRA [14] tuned)
	VideoLLM-online [9]	Use all	$1 \rightarrow 0.6k$
	VIDEOLLM-MOD	Use all	$1 + 3 \times 3 \rightarrow 6k$

Figure 7: Vision strategy in popular LMMs.

introduce a simple data augmentation strategy that requires no additional data annotation but, in our observation, significantly alleviates overfitting. The core idea is to disrupt the language context during streaming video-language modeling, reducing the model's bias towards language alone. Our "disruption" method involves randomly shifting within a temporal window and randomly replacing text with text from other timestamps within the same video data, or both. For the disrupted text, we do not perform language modeling or streaming EOS modeling on its adjacent video frames.

All figures presented in this paper were obtained using this augmentation method. We will further investigate whether this approach can mitigate the shortage of high-quality streaming video-language data.

19

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We describe our method and contributions precisely.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We detailed our limitations in the conclusion section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

20

Justification: This work focuses on the application of video understanding and does not include theoretical results or analysis.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We clearly illustrate all details and experimental settings of our approach, which rely on open-source datasets and models.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

21

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We will upload our code as an attachment.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Our work primarily uses public benchmarks for experiments, with all experimental settings detailed in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Error bars are not reported due to the high computational expense involved for all settings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

22

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide the machine specifications and experimental details in the experiment section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Our work complies with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: In the introduction and conclusion sections, we discuss how our work could be applied to real-world video applications such as an online assistant.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

23

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: We do not collect data ourselves; instead, we rely on the original safeguards of the datasets and models we use.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use open-source datasets such as Ego4D, EgoExo4D, and COIN, and have obtained official licenses for them.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

24

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide comprehensive documentation for all new assets introduced in the paper.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We don't do crowdsourcing and research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

25

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

26