

Streaming VideoLLMs for Real-time Procedural Video Understanding

Dibyadip Chatterjee^{1,3*} Edoardo Remelli¹ Yale Song² Bugra Tekin¹ Abhay Mittal¹ Bharat Bhatnagar¹
 Necati Cihan Camgöz¹ Shreyas Hampali¹ Eric Sauser¹ Shugao Ma¹ Angela Yao³ Fadime Sener¹

¹ Meta Reality Labs ² FAIR, Meta ³ National University of Singapore

dibschat.github.io/ProVideLLM

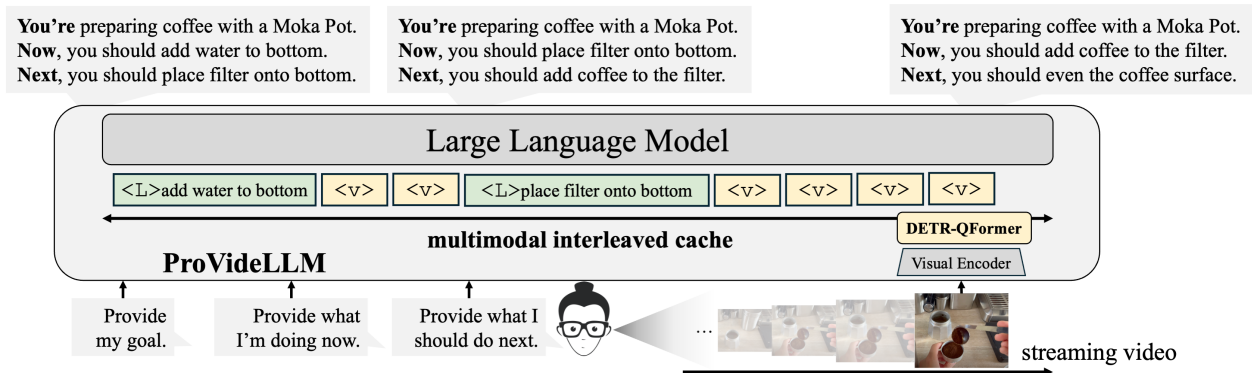


Figure 1. **Overview of ProVideLLM.** A streaming video large language model for real-time procedural video tasks with a low memory footprint. It features a multimodal interleaved cache that stores textual descriptions of long-term observations (spanning several minutes) and visual tokens representing short-term observations (spanning a few seconds). We also introduce a new DETR-QFormer connector for better fine-grained tokenization of the short-term. ProVideLLM is capable of handling multiple procedural tasks within a single model.

Abstract

We introduce *ProVideLLM*, an end-to-end framework for real-time procedural video understanding. *ProVideLLM* integrates a multimodal cache configured to store two types of tokens – verbalized text tokens, which provide compressed textual summaries of long-term observations, and visual tokens, encoded with *DETR-QFormer* to capture fine-grained details from short-term observations. This design reduces token count by $22\times$ over existing methods when representing one hour of long-term observations while effectively encoding fine-granularity of the present. By interleaving these tokens in our multimodal cache, *ProVideLLM* achieves sub-linear scaling of memory and compute with video length, ensuring per-frame streaming inference at 10 FPS and streaming dialogue at 25 FPS, with a minimal 2GB GPU memory footprint. *ProVideLLM* also sets new state-of-the-art results on six procedural tasks across four datasets.

*Work done during an internship at Meta.

1. Introduction

Procedural videos, spanning several minutes to hours, capture multi-step tasks [82] such as cooking [65, 101], toy assembly [60, 67], and furniture building [7, 8]. With the growing adoption of AR assistants, streaming video understanding is essential for real-time procedural assistance. This requires real-time, online inference, as frames appear in a live video, without access to future observations. Despite its importance, streaming video understanding remains largely unexplored in existing procedural research [19, 36, 100].

Existing works on streaming video understanding [17, 70] focus on online action detection [53, 79, 86, 97] or live conversations [14, 83]. These use-cases target a small set of unrelated actions [17, 34] and do not require long-term temporal modeling. In contrast, procedural videos are inherently long and causal, where each step depends on prior steps. Applying existing approaches to such videos significantly increases token count and memory overhead, leading to long-context degradation [31, 46] and making them unsuitable for streaming, even with excessive frame dropping [29]. The semantic inter-connectivity of activity steps introduces re-

dundancy when long-term observations are stored as raw frame tokens [79, 86]. Furthermore, while short-term observations are crucial for understanding the present, not all frame tokens are equally relevant, since procedural tasks often center around hand-object interactions [28, 67, 72].

In this paper, we propose an end-to-end framework for streaming procedural video understanding (Fig. 1), addressing the above mentioned challenges of processing long-form videos. Specifically, we introduce ProVideLLM, a Procedural VideoLLM that comprises a visual (frame) encoder, a vision-language connector, and a pre-trained large language model (LLM) [1, 20, 58, 77] as the decoder. Our key contribution is a vision-language multimodal cache that efficiently manages streaming video data by balancing compression and representation quality of past observations.

We factorize the streaming past observations into long-term semantic history and short-term visual context. They are encoded as text and visual tokens, respectively, and interleaved in our multimodal cache. First, for the long-term history spanning several minutes, ProVideLLM decoder predicts step labels per frame, groups similar steps into verbalized sequences (e.g. *get* → *crack* → *whisk eggs*), and stores them in the cache. This semantic compression significantly reduces token count and also enhances current step prediction (e.g. → *pour mixture*). Second, to capture short-term visual context, typically spanning a few seconds, understanding the fine-grained changes across frames is crucial. However, standard VideoLLM encoders [59, 89] face two key challenges in such modeling: (i) temporal collapse, where low temporal variance conflates distinct actions and (ii) background distractors across frames [5, 13, 61, 90]. We mitigate the former by using visual encoders trained with dense spatial alignment objectives [11, 30, 52]. We address the latter with a novel vision-text connector DETR-QFormer, pretrained to explicitly focus on hand-object interaction regions, thereby compressing them into a smaller set of tokens.

Finally, in existing methods [86], both long- and short-term tokens are visual and stored separately. Applying such separate caching to our multimodal tokens would require redundant attention recomputation when verbalizing short-term visual tokens into long-term text. To overcome this, we interleave text and visual tokens within our multimodal cache. This enables prefilling the cache, which ensures that computation scales linearly with cache size and sub-linearly with the video length, thereby significantly reducing latency for real-time performance.

In summary, our contributions are: (i) a first-of-its-kind streaming multimodal cache with interleaved text and visual tokens, enabling per-frame inference [72] at 10 FPS and streaming dialogue [14] at 25 FPS; (ii) verbalizing long-term observations into compressed textual summaries and (iii) modeling hand-object interactions for short-term observations using DETR-QFormer; all combined, it achieves a

22× reduction in token count compared to existing methods. (iv) state-of-the-art results on six tasks across four datasets, making ProVideLLM the first real-time framework unifying recognition, anticipation, planning for procedural guidance.

2. Related Works

Procedural Video Understanding have primarily focused on instructional YouTube videos [48, 74, 101, 103]. Recent large-scale egocentric datasets [27, 28, 67, 72] have advanced research by providing longer videos with more activity steps. Existing works have focused on offline tasks [19, 43, 100], where the entire video is available for prediction, as well as some online tasks, like procedural planning [12, 33, 95], forecasting [23, 26, 49, 65, 66, 96], and mistake detection [18, 22, 37, 67]. However, deploying these models for streaming inference results in high computational complexity where it grows quadratically with the video length. In contrast, ProVideLLM incurs a sub-linear cost per update, both in terms of memory and runtime, enabling efficient streaming inference.

Video Large Language Models. Multimodal large language models (MLLMs) have been popularized by BLIP-2-like [15, 39, 102] and LLaVA-like [38, 44, 45] models where an LLM is trained to understand vision-text data. Extending these to video have led to the development of Video Large Language Models (VideoLLMs) [40, 41, 47, 83, 94, 98]. However, most of these models are limited to offline tasks [62, 71, 73, 80, 81] where the entire video is queried to answer a question. As such, architectural elements such as VideoQ-Former [92] and temporal pooling layers [40], developed without causal requirements, limit streaming applications. Models like MA-LMM [29] and VideoLLM-online [14] address video understanding *online*, but are too computationally intensive to be deployed for efficient streaming inference on long-form videos. ProVideLLM advances these models by introducing a multimodal interleaved cache for faster inference and proposing DETR-QFormer for modeling fine-grained visual changes, improving accuracy.

Streaming Video Understanding has been explored for online action detection [17, 25, 70] where dominant approaches [53, 79, 86, 97] split streaming input into long- and short-term visual histories within an encoder-decoder architecture. However, storing both as visual tokens is redundant and computationally prohibitive for long-form procedural videos, where multiple actions are temporally connected. Instead, ProVideLLM verbalizes long-term observations into text which drastically decreases token count required to represent hour long videos, thus improving memory efficiency.

3. ProVideLLM

ProVideLLM encodes long- and short-term streaming observations into text and visual tokens and interleaves them in a multimodal cache. We first present the preliminaries

in Sec. 3.1, followed by long- and short-term token generation in Sec. 3.2 and Sec. 3.3. Finally, we present our multimodal cache in Sec. 3.4 and training objectives in Sec. 3.5.

3.1. Preliminaries

ProVideLLM follows a LLaVA-style [44] architecture, consisting of a visual encoder (E_V), a language decoder (D_L), and a vision-language connector (C_{VL}). Given a video $V = \{v_1, v_2, \dots, v_T\}$, the model’s output (o_t) at any time step $t \in [1, T]$ is

$$o_t = \text{ProVideLLM}(v_{1:t}) = D_L \left(\left\|_{i=1}^t C_{VL}(E_V(v_i)) \right\| \right) \quad (1)$$

where $\|$ denotes concatenation. While LSTR-inspired long short-term memory approaches [79, 97] also use an encoder-decoder architecture, they rely exclusively on visual tokens (Fig. 2a.1). In contrast, ProVideLLM incorporates a language decoder, which enables more efficient long-term compression, improves generalization, and supports multiple tasks within a single model.

3.2. Tokenizing long-term observations

Procedural videos are inherently long-form, ranging from tens of minutes [67] to several hours [28, 72]. While modern LLMs [20, 75, 87] offer massive context lengths and, in theory, can process an hour-long raw video at 24 fps, performance degrades with larger context sizes [31, 46], and training such models is highly memory-intensive [29, 93]. Accurately recognizing ongoing steps require effectively encoding long-term context. Our key insight is that procedural steps in a video are temporally connected, *i.e.* a sequence of step labels can serve as a strong predictor of ongoing steps. For instance, when “*preparing an omelet*”, the step “*pour mixture into pan*” depends on earlier steps “*get eggs from fridge, crack eggs into bowl, whisk eggs*”. Prior works on flow graphs [21, 64] and planning [3, 54, 100] also show that step sequences, often extracted from external sources [35], serve as strong baselines for procedural understanding.

Instead of constructing computationally expensive graphs, we propose *online verbalization* (Fig. 2a.2). Given a streaming video, ProVideLLM decoder continuously generates action predictions (summaries) of the present observation on the fly, and groups semantically similar actions to verbalize the past (Fig. 1). This approach efficiently compresses past visual frame tokens into information-dense language tokens. For example, in Ego4D Goal-Step [72], storing one hour of long-term past requires only 630 verbalized text tokens on average. Compared to long short-term approaches [79, 86, 97], our verbalization reduces token count by $22\times$ at the same sampling rate, significantly reducing the memory overhead.

3.3. Tokenizing short-term observations

Our goal is to recognize procedural activity steps in a streaming manner. This requires understanding “what is happening” in the present while tracking “what has happened in

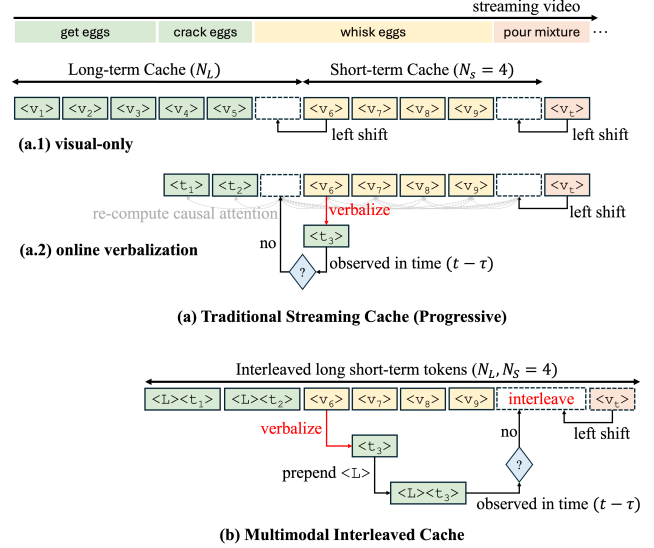


Figure 2. **Converting short-term tokens to long-term under various caching types.** (a) Factorizing the streaming observations into long-term and short-term and ordering them progressively. (a.1) Conversion done at $O(1)$ cost but is unable to represent long-form videos due to massive token count and memory needs. (a.2) Online verbalization can represent long-form videos at a manageable memory but is not suitable for streaming due to $O(N_S^2 + N_S N_L)$ conversion cost. (b) Interleaving them reduces the conversion cost to $O(N)$, allowing streaming inference on long-form videos.

the past”. Language is a strong indicator of the semantic structure of step sequences, but determining if and when the step is performed requires fine-grained visual information. Therefore, we allocate more compute for understanding the short-term past, typically spanning 8-16 seconds, by keeping its visual tokens uncompressed. Standard VideoLLM encoders [38, 44] face two key challenges in capturing fine-grained short-term observations: (i) they generate temporally collapsed video features, a limitation we identify and address in Sec. 3.3.1, and (ii) they get biased towards high-level scene and background distractors, which we address in Sec. 3.3.2.

3.3.1. Choosing the visual encoder

The predominant visual encoders for MLLMs are CLIP [59] and SigLIP [2, 89]. These encoders are pre-trained with only a language-alignment objective for integration with LLMs. However, because they are trained on high-level and sparse web-scale captions such as “*cooking in the kitchen*”, they struggle with fine-grained reasoning, particularly in answering the “*what?*” and “*how to?*” aspects of a video [84, 85]. Consequently, distinct images may yield high similarity scores in their feature space [76], leading to low feature diversity across frames within a step. This results in relatively static frame features, a phenomenon we term *temporal collapse*. For certain fine-grained procedural steps, such as “*pick up*” vs. “*put down*”, temporally collapsed

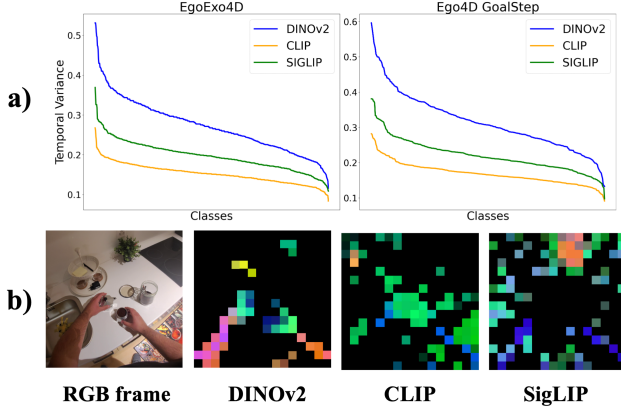


Figure 3. **a) Per-class mean temporal variance** on EgoExo4D & Ego4D Goal-Step. Highly temporally varying classes on the left and low-variance classes on the right. **b) Visualization of the top-3 PCA components of patch tokens** computed across video frames, thresholded by first component. The primary factors of variation for DINOv2 [52] are hands, and are unfocused and scattered across the image for language-aligned encoders, CLIP [59] and SigLIP [89].

features make it nearly impossible for the downstream LLM to decode the correct step.

Fig. 3a shows the impact of temporal collapse across three visual encoders – two language-aligned ones [59, 89] and one self-supervised¹ [52]. We plot temporal variance per class in descending order, with high-variance classes on the left and low-variance classes on the right (see Suppl.Sec.1 for temporal variance computation). At both extremes, the trends are consistent across all encoders. High-variance classes, which often involve verbs such as “get”, “fetch” or “put” correspond to longer videos with substantial scene and environmental changes. Conversely, low-variance classes involving adverbs such as “slowly” or “boil” or repetitive actions like “rotate” or “peel”, exhibit gradual state changes. Among the three models, DINOv2 consistently captures temporal variations more effectively, covering a broader range of temporal variance across different classes. Hence, we choose DINOv2 as our visual encoder.

3.3.2. Aligning hand-object transformations with text

Procedural datasets predominantly capture hand-centric actions. For instance, nearly 99% of actions in [28, 72] involve hand interactions². Moreover, video frames often contain distracting background objects, while hand locations serve as strong cues for recognizing the ongoing actions [68, 91]. Thus, a vision encoder focused on hand regions can enhance performance by mitigating background distractions and capturing fine-grained transformations more precisely.

Another contribution of our work is a vision-language connector, DETR-QFormer, pretrained to attend to salient

¹We chose DINOv2 due to its comparable scale to CLIP and SigLIP.

²Remaining 1% are visually undetectable actions “call for help, waiting.”

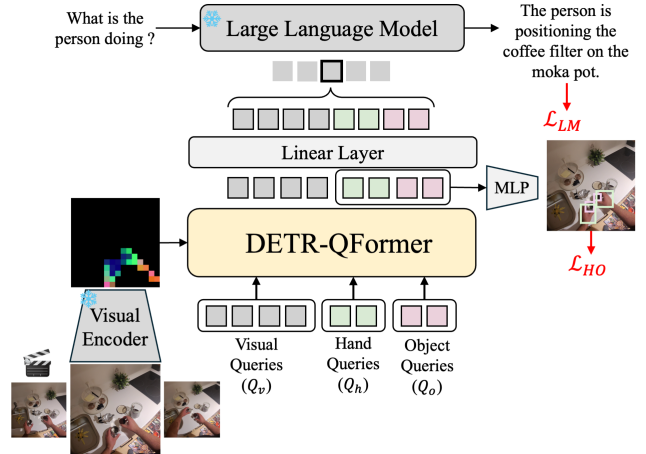


Figure 4. **DETR-QFormer architecture & Stage-1 pre-training.** Trained to detect hands and objects-in-contact from pseudo ground truths bboxes generated by [69], DETR-QFormer queries DINOv2 patch tokens to extract hand-object-focused activations.

objects. In procedural videos, these are typically in contact with the hand. Interestingly, our choice of DINOv2 as the visual encoder provides an advantage as its patch tokens capture hands as primary factors of variation Fig. 3b. Leveraging this, we design DETR-QFormer to query DINOv2 patch tokens, extracting hand-object activations and suppressing background distractions. DETR-QFormer explicitly attends to hand-object activations using a DETR loss [10] and compresses salient regions into fewer tokens via QFormer’s [39] cross-attention, enhancing representation quality.

Architecture. Our connector integrates Q-Former’s compression [39] and DETR’s bbox loss into a single cross-attention block. DETR-QFormer, illustrated in Fig. 4, is pretrained to detect the hands and objects-in-contact (Sec. 3.5). The connector (C_{VL}) is a cross-attention transformer decoder with a set of m learnable visual queries $Q_v = (q_{v_1}, q_{v_2}, \dots, q_{v_m})$. Key-value pairs (K, V) are generated from the flattened n^2 DINOv2 patch tokens using two projection heads (W_K, W_V) respectively. To explicitly model hand-object interactions, we append two sets of special queries – two hand queries $Q_h = (q_{h_1}, q_{h_2})$ for the left and right hands, and k object queries $Q_o = (q_{o_1}, q_{o_2}, \dots, q_{o_k})$ corresponding to objects-in-contact. The role of visual queries Q_v is to compress the $n^2 \times D$ patch tokens into $m \times d$ tokens, where $m \ll n^2$ and $d \ll D$. Whereas, outputs corresponding to the learnable hand-object queries (Q'_h, Q'_o) are passed through an MLP to make bounding box predictions, $\hat{B} = \{\hat{b}_{h_1}, \hat{b}_{h_2}, \hat{b}_{o_1}, \dots, \hat{b}_{o_k}\}$.

3.4. Multimodal Interleaved Cache

The effectiveness of a procedural video model is best assessed when deployed for real-time inference. Naive long- and short-term approaches [79, 86] maintain separate caches

Algorithm 1 Multimodal Interleaved Cache

```
1: class MultimodalInterleavedCache:
2:   def __init__(self, N_S, N_L):
3:     self.tokens = []
4:     self.N_S = N_S
5:     self.N_L = N_L
6:   # entry function: adds a new token to cache
7:   def entry(self, token):
8:     self.tokens.append(token)
9:   # exit function for short-term tokens
10:  def exit_short(self):
11:    if self.tokens.count(<V>) > self.N_S:
12:      self.tokens.pop(self.tokens.index(<V>))
13:      return True
14:  # exit function for long-term tokens
15:  def exit_long(self):
16:    if self.tokens.count(<L>) > self.N_L:
17:      self.tokens.pop(self.tokens.index(<L>))
```

for the two token types, arranged progressively with a single entry point (see Fig. 2a). With multimodal tokens, such setups require two distinct entry points, forcing attention recomputation over the short-term cache whenever a short-term token is verbalized into a long-term one. In the worst case, this leads to quadratic scaling of compute. To efficiently leverage online verbalization, we propose a *multimodal interleaved cache* that interleaves long-term text and short-term visual tokens from Sec. 3.2 and Sec. 3.3.

Interleaving long- and short-term tokens. We store the running short-term visual tokens and long-term text tokens in a single multimodal cache operating as a FIFO (First In, First Out) queue. It has one entry point and two distinct exits for short- and long-term tokens. Our entry and exit functions are detailed in Algorithm 1, where N_S denotes the short-term cache length (in number of frames) and N_L the long-term cache length (in number of observed steps). As frame tokens (v_i) exit the short-term span, the ProVideLLM decoder verbalizes the oldest token (v_{-N_S}). The verbalized text is then added to the cache unless already cached within the past τ frames. To distinguish the two token types within the same cache, we prepend a special token (represented as $\langle L \rangle$) to the long-term text tokens. ProVideLLM is instruction-tuned to recognize tokens associated with $\langle L \rangle$ as belonging to the long-term observations (Sec. 3.5). Since both long- and short-term tokens share a single entry point, visual and textual tokens are interleaved in the cache at any given time.

Streaming inference. Every time a new token is added to the cache (Algorithm 2), instead of re-computing attention across all tokens, we use the causal decoder’s K-V cache [55] to compute attention only for the new token over the previous $N - 1$ tokens. This ensures that the computational complexity for processing a new frame scales linearly with cache size ($O(N)$). Additionally, maintaining a single interleaved cache enables direct application of optimized attention implementations [6, 81], though such techniques remain orthogonal to our approach. More importantly, for a fixed N_S , compression via verbalization ensures that N_L

Algorithm 2 Streaming Inference

```
1: def inference(V, N_S, N_L,  $\tau$ ):
2:   cache = MultimodalInterleavedCache(N_S, N_L,  $\tau$ )
3:   out = []
4:   for  $v_i$  in V:
5:     cache.entry( $v_i$ ) # add frame tokens to cache
6:      $t_i$  = ProVideLLM(cache.tokens) # current prediction
7:     out.append( $t_i$ ) # buffer predictions
8:     flag = cache.exit_short() # short-term overflow
9:     if flag and out[-N_S] not in out[- $\tau$ ]:
10:      cache.entry(out[-N_S]) # verbalize & interleave
11:      cache.exit_long() # long-term overflow
```

and consequently $N = (N_L + N_S)$ scales sub-linearly with the number of video frames. Hence, compute also scales sub-linearly with the number of streaming video frames. This makes our model well-suited for low-latency applications.

3.5. Training ProVideLLM

We adopt a two-stage training approach similar to LLaVA – (1) alignment pre-training to align the encoder’s visual tokens to the decoder’s language space, and (2) instruction tuning of the model for specific tasks.

Stage-1: Alignment Pre-training. In this stage, our connector DETR-QFormer (C_{VL}) is trained while keeping the visual encoder and language decoder frozen (Fig. 4). The objective is to caption short video clips featuring hand-centric actions using the standard language modeling loss [57]:

$$\mathcal{L}_{LM} = -\frac{1}{|W|} \sum_{i=1}^{|W|} \log P(w_i | w_{<i>}, V; C_{VL}^\Theta), \quad (2)$$

where V is the input video clip, $W = \{w_1, w_2, \dots, w_{|C|}\}$ is the ground-truth caption, and C_{VL}^Θ is our DETR-QFormer parameterized by Θ . For detecting hands and corresponding objects-in-contact, we use a DETR [10]-style Generalized IoU [63] loss:

$$\mathcal{L}_{HO}(b_i, \hat{b}_{\sigma(i)}) = \mathcal{L}_{iou}(b_i, \hat{b}_{\sigma(i)}) + \|b_i - \hat{b}_{\sigma(i)}\|_1 \quad (3)$$

where b_i ’s are the pseudo ground-truth bounding boxes of hands and objects-in-contact generated by an off-the-shelf detector [69] and $\hat{b}_{\sigma(i)} \in \hat{B}$ is the bounding box matched from the predicted set \hat{B} . For simplicity, we keep our loss category-agnostic. The final loss for Stage-1 is thus:

$$\mathcal{L} = \mathcal{L}_{LM} + \lambda_1 \mathcal{L}_{HO} \quad (4)$$

where λ_1 is a hyper-parameter. In addition to vision-language alignment, Stage-1 also enables our model to attend to hand-object interactions.

Stage-2: Instruction Tuning. This stage focuses on fine-tuning our model with task-specific instructions such as “Provide the goal.”, “Provide what I am doing now.”, “Provide what I need to do next.”, or “Provide the

Model	per-frame mAP	
	Val	Test
LSTR [86] (only short-term)	8.8	-
LSTR [86]	8.9	8.1
EgoOnly [78]	10.3	10.9
ProVideLLM-1B/5 (no verbalization)	<u>12.1</u>	<u>12.2</u>
ProVideLLM-1B/5 (verbalization & interleaving)	13.0	12.9

Table 1. **Ego4D [72] GoalStep Online Step Detection.** For fair comparison with LSTR, we consider 16s of short-term past and an avg. of 128s of long-term past compressed into language tokens.

Model	Views	Action
TempAgg [66]	12 views	8.5
+ Goal [88]	12 views	<u>12.0</u>
ProVideLLM-8B/11 (+long-term verbalization)	v4	13.8

Table 2. **Action Anticipation Results on Assembly101** validation set. Performance measured by class-mean Top-5 recall.

next 10 steps I need to do.”. For instruction tuning, we fine-tune DETR-QFormer and the language decoder using \mathcal{L}_{LM} , while keeping the visual encoder frozen. For streaming tasks, we also train our model with interleaved special $\langle L \rangle$ tokens to represent long-term cache entries, distinguishing them from standard prompts. See Suppl.Sec.5 for further details.

4. Experiments

We conduct extensive evaluations of ProVideLLM on six tasks across four datasets covering streaming (Sec. 4.1.1), long-term (Sec. 4.1.2) and fine-grained (Sec. 4.1.3) procedural understanding. We further assess ProVideLLM’s performance in multi-task settings and its generalization to new datasets in Sec. 4.2. Finally, we provide ablations in Sec. 4.3 and ProVideLLM’s runtime analysis in Sec. 4.4.

Implementation Details. Depending on the choice of vision encoder and language decoder, we have two model variants. ProVideLLM-1B/5 is our efficiency-focused model designed for real-time streaming. It uses DINOv2 (ViT-S) with Llama-3.2-1B-Instruct [20] and requires 5 tokens per frame. ProVideLLM-8B/11 is our more performance-focused model using DINOv2 (ViT-g) and Llama-3.1-8B-Instruct and requires 11 tokens per frame. Furthermore, for benchmarking on youtube videos [74], we introduce an additional model ProVideLLM-8B/11+, which uses SigLIP’s [CLS] token along with DINOv2 patch tokens. Unless otherwise mentioned we use $N_S = 64, N_L = 5$ *i.e.* storing a maximum of 64 frames in the short-term cache and a maximum of 5 observed steps in the long-term cache. DETR-QFormer is pre-trained (Stage-1) on EPIC-KITCHENS [16] with pseudo ground-truth bounding boxes from 100DOH [69]. For Stage-2, the LLM (decoder) is fine-tuned using LoRA [32] with $r=128$ and $\alpha=256$. Additional implementation details are provided in Suppl.Sec.5.

Method	Top-1 Acc. (%)
TimeSformer [9]	34.0
DistantSup	39.4
VideoTF	42.4
ProcedureVRL	46.8
VEDIT [42]	51.8
VideoLLM-online [14]	49.1
VideoLLM-MoD [83]	49.7
ProVideLLM-8B/11+	<u>52.1</u>
ProVideLLM-8B/11+ (+long-term verbalization)	53.6

Table 3. **Next Step Forecasting in COIN.** The last four rows including ProVideLLM are VideoLLMs.

Datasets. We evaluate our model’s performance on four large-scale procedural datasets – EgoExo4D [28], Ego4D-GoalStep [72], COIN [74], and Assembly101 [67]. As shown in Sec. 4.1, ProVideLLM sets state-of-the-art on both egocentric and YouTube benchmarks, demonstrating robustness across diverse video domains.

4.1. Main results

4.1.1. Online Step Detection

We report results in Tab. 1, using per-frame mAP on action classes (steps + substeps) at 4fps, a standard metric in online action detection [79, 97]. We evaluate ProVideLLM-1B/5 with and without long-term verbalization in rows four and five, respectively. Results for LSTR [86] and EgoOnly [78] are taken from [72]. Accordingly, LSTR spans 128s in long-term memory and 16s in short-term memory. In [72], each step spans on average 32s with around 5.7 tokens representing each step. Our long-term cache can encode 128s of past using only 22 tokens at $N_L = 4$ steps. To measure improvements from our long-term design, we set the cache length $N = 86$ (spanning LSTR’s temporal context), split into ($N_S = 86, N_L = 0$) and ($N_S = 64, N_L = 4$) in rows four and five, respectively. Our model surpasses both baselines, outperforming LSTR by over 4% at similar temporal span. ProVideLLM also achieves greater gains from long-term context compared to LSTR, highlighting the effectiveness of verbalizing the long-term past.

4.1.2. Online Step Forecasting

Step forecasting requires modeling long-term dependencies, making it an ideal task to evaluate our contributions. We evaluate ProVideLLM for step forecasting on COIN and Assembly101 measured by Top-1 accuracy and Top-5 Recall [24], respectively. Our model achieves state-of-the-art performance on both datasets as shown in Tab. 3 and Tab. 2. On Assembly101, we surpass [88] by 1.8% using only a single view. On COIN, we use Llama-3-8B-Instruct to ensure fair comparisons with VideoLLMs [14, 83] and achieve a performance boost of around 4%. Moreover, while state-of-the-art models [14, 83] use approximately 600 to 6000 visual

Model	Training Views	Ego Acc. (%) Val / Test
TimeSFormer [9]	ego	35.13 / 35.93
EgoVLPv2 [56]	ego+exo	39.10 / 38.76
Viewpoint Distillation	ego+exo	38.19 / 39.49
View Invariant Encoder [51]	ego+exo	40.34 / 41.53
VideoLLM-MoD [†] [83]	ego	42.62 / —
ProVideLLM-8B/11	ego	44.36 / 50.74

Table 4. **EgoExo4D Fine-grained Keystep Recognition.** Test results obtained from EgoExo4D’s Fine-grained Keystep Recognition Challenge on EvalAI. [†] Uses 2fps sampling per segment.

Method	Top-1 Accuracy (%)
TimeSformer [9]	34.0
Paprika [100]	51.0
DistantSup [43]	54.1
ProcedureVRL [99]	56.9
VideoTaskGraph [4]	57.2
VEDIT [42]	62.7
VideoLLM-online [14]	63.1
VideoLLM-MoD [83]	63.4
ProVideLLM-8B/11+	67.3

Table 5. **COIN Step Recognition.** Last three rows are VideoLLMs.

tokens to model past context, our approach spans a larger temporal context with only 200 tokens through verbalization.

4.1.3. Step Recognition

In Tab. 4 and Tab. 5, we evaluate ProVideLLM on fine-grained step recognition on EgoExo4D and COIN. We uniformly sample 16 frames from each segment to construct our short-term cache (N_S) and use the prediction from the last frame as our output. The closest model, VideoLLM-MoD [83], uses 10 tokens per frame, but densely samples frames at 2fps, resulting in a larger token count. In both cases, we outperform VideoLLM-MoD by 1.74% and 3.9% respectively. Moreover, at the time of our submission, our method, using only egocentric video, achieved the best results on the EgoExo4D Fine-grained Recognition Challenge, with a 9.2% improvement over the second-ranked model.

4.2. Generalized Procedural Understanding

In this section, we explore additional capabilities of ProVideLLM, leveraging the advantages of using a pre-trained LLM as the language decoder. This section provides insights into developing a “*generalist*” model that advance procedural video understanding within a unified framework. We first evaluate ProVideLLM’s ability to multi-task and then present our benchmark for cross dataset generalization. **Multi-task learning.** In Tab. 6, we present our results for multi-task learning, an underexplored aspect of procedural understanding. Existing models are trained only for isolated tasks, such as step recognition and forecasting. We compare

Model	COIN Benchmark Top-1 Accuracy (%)				
	Step	Task	Next	Proc.	Proc.+
VideoTF [†] [50]	56.5	91.0	42.4	40.2	46.4
VideoLLM-online [14]	63.1	92.7	49.1	49.8	<u>54.1</u>
VideoLLM-MoD [83]	63.4	92.8	49.7	49.8	53.3
ProVideLLM-8B/11+	66.9	95.0	50.5	51.0	55.9

Table 6. **Multi-task learning on COIN.** Step – step recognition, Task – task recognition, Next – next step forecasting, Proc. – long-term forecasting and Proc+ – long-term forecasting with task. [†]VideoTF [50] is trained separately for each task.

Model	Ego Acc. (%)
VideoLLaVA [41] + Llama-3.1-8B [20]	3.6
LLaVA-OneVision [38] + Llama-3.1-8B [20]	8.3
ProVideLLM-8B/11	12.4

Table 7. **Cross-dataset generalization.** Evaluated on EgoExo4D KeyStep recognition. Our model, trained on Ego4D Goal-Step is compared with a mixture of vision + language experts.

our model with recent VideoLLMs, also trained in a multi-task setting, on 5 tasks. Despite using the same LLM as [14, 83], our model outperforms these baselines across all tasks. **Cross-Dataset Generalization.** In Tab. 7, we present a cross-dataset generalization benchmark for procedural videos using the EgoExo4D fine-grained keystep recognition task. We compare two recent VideoLLMs [38, 41] which were primarily trained for video captioning and use Llama-3.1-8B-Instruct to convert the captions into EgoExo4D classes. Our baseline models act as a combination of visual and language experts. ProVideLLM, using the same LLM and trained on the much smaller Ego4D Goal-Step dataset, significantly outperforms these baselines. Despite these advances, performance remains modest, highlighting the need for further research in procedural generalization. Additional details are provided in the Suppl. Material.

4.3. Ablations

Visual Encoders & DETR-QFormer. In Tab. 8, we ablate visual encoders—SigLIP and DINOv2—and connectors—MLP, QFormer, and our DETR-QFormer—on EgoExo4D keystep recognition. First, comparing encoders across connectors, SigLIP’s CLS tokens, being language-aligned, outperform DINOv2; however, with more patch tokens, DINOv2 surpasses SigLIP, aligning with our observations in Fig. 3. Second, comparing connectors across encoders, Q-Former boosts DINOv2’s performance (+1.2%, +1.9%) but has minimal impact on SigLIP (+0.8%, -0.1%), highlighting DINOv2’s need for a stronger connector. Finally, DETR-QFormer further improves DINOv2 (+1.1%, +2.6%), with gains increasing at higher resolutions but degrading SigLIP’s performance due to weak hand-object activations in its patch tokens. Notably, DETR-QFormer remains highly

[CLS] Token	Patch Tokens	#Tokens /frame	Ego Val./ Acc. (%)	
			SigLIP	DINOv2
w/ 2-layer MLP [45]				
✓	✗	1	31.8±0.4	28.7±0.2
✓	16×16 → 2×2	5	32.1±0.6	32.4±0.5
✓	16×16 → 4×4	17	33.3±0.6	36.6±0.5
w/ QFormer [39]				
✓	16×16 ⇒ 4	5	32.9±0.6	33.6±0.5
✓	16×16 ⇒ 16	17	33.2±0.4	38.1±0.7
w/ DETR-QFormer (2 hands, 2 objects)				
✓	16×16 ⇒ 0	5	30.9±0.3	34.7±0.5
✓	16×16 ⇒ 12	17	33.6±0.3	40.7±0.6

Table 8. **DETR-QFormer ablations** on EgoExo4D fine-grained keysteps. Llama-3.2-1B-Instruct is used as the language decoder. Results reported over 3 runs. (→ means avg-pooling, ⇒ means Q-Former compression.)

token-efficient with DINOv2, even when compressing to only hand-object queries, reinforcing its advantage for low-latency streaming applications.

Verbalization & Interleaving. In Fig. 5, we analyze our verbalization and interleaving strategies using ProVideLLM-1B/5 on Ego4D Goal-Step validation set, selecting a video containing over 15 steps and spanning >30 minutes. At 4fps sampling, we ablate on a fixed memory budget of 8GB on an A6000 GPU. Short-term cache length is set as $N_S = 64$ (16 seconds), and no limit is imposed on the long-term cache. Using only visual tokens (red), memory scales linearly with video length, reaching 40K tokens to represent 20 minutes of video before exceeding 8GB memory. Such massive context also degrades performance. Verbalization (orange) compresses 20 minutes of video into 2.5K tokens, reducing memory usage to 3GB and achieving sub-linear scaling. However, it introduces runtime instability, with frequent spikes in per-frame prediction time due to attention recomputation when converting short-term visual tokens into long-term text. Interleaving (green) in our multimodal cache mitigates these spikes, improving runtime stability while maintaining sub-linear scaling in both memory and compute.

4.4. Runtime Analysis

In Tab. 9, we report the per-frame streaming inference and streaming dialogue [14] performance of ProVideLLM variants on a single A6000 GPU. Our lightest variant, ProVideLLM-1B/5, achieves over 10 FPS with a minimal 2GB memory footprint, making it ideal for low-latency applications. Meanwhile, ProVideLLM-8B/11 offer higher performance but require more compute. For most videos in this dataset, ProVideLLM-1B/5 consistently maintains over 10 FPS, sometimes even reaching 20 FPS when the video has many no-action frames. For per-frame predictions, a key bottleneck for VideoLLMs is their autoregressive text gener-

Streaming Predictions	Model	FPS (↑)		Memory (GB) (↓)	
		Vision Connector		Full Model	
		Enc.	+ LLM		
Per-frame	ProVideLLM-1B/5	74.8	10.4	9.1	2.0
	ProVideLLM-8B/11	38.0	4.2	3.5	16.2
Dialogue [14]	ProVideLLM-1B/5	74.8	34.2	24.6	2.2
	ProVideLLM-8B/11	38.0	23.7	17.2	16.9

Table 9. **Runtime Analysis** on a single A6000 GPU for the validation set of Ego4D-GoalStep online step detection for per-frame streaming and Ego4D-GoalStep Narration [83].

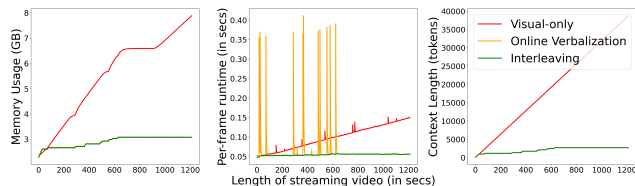


Figure 5. **Scaling of memory, runtime and context length under various caching strategies.** Results reported on a Ego4D-GoalStep video at a memory budget of 8GB on a single A6000 GPU.

ation, adding around 6 extra iterations, corresponding to 5.7 tokens/(activity step) on average for Ego4D-GoalStep. This issue is mitigated with streaming dialogue [14], where the model is trained to remain silent and respond to user queries only when necessary. In this setting, ProVideLLM-1B/5 delivers real-time performance at 24.6 FPS.

5. Conclusion and Limitations

In this paper, we presented ProVideLLM for real-time understanding of procedural videos by interleaving two types of tokens in a vision-text multimodal cache – verbalized text tokens, which provide compressed summaries of long-term observations, and visual tokens, which capture fine-grained details from short-term observations. We show that ProVideLLM reduces the token count for representing long-form procedural videos while effectively capturing critical spatio-temporal dependencies. We demonstrated state-of-the-art results on six tasks across four datasets, showing per-frame inference at 10 FPS and streaming dialogue at 25 FPS. ProVideLLM is the first real-time framework unifying recognition, anticipation, and planning for procedural assistance. Our multimodal interleaved cache is also generic, applicable to non-procedural videos, and unlocking a new area of compressing long videos on the fly while reducing redundancy and boosting performance.

Limitations. DETR-QFormer, trained on pseudo ground-truth bounding boxes, generalizes well but could benefit from GT data to reduce biases from noisy annotations. Additionally ProVideLLM focuses on detection but lacks real-time feedback on the “how” of task execution. Finally, as discussed in Sec. 4.2, ProVideLLM’s generalization performance remains modest, presenting a potential direction for future work.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 2
- [2] Ibrahim M Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. Getting vit in shape: Scaling laws for compute-optimal model design. *Advances in Neural Information Processing Systems*, 36, 2023. 3
- [3] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystone recognition in instructional videos. *Advances in Neural Information Processing Systems*, 36: 67833–67846, 2023. 3
- [4] Kumar Ashutosh, Santhosh Kumar Ramakrishnan, Triantafyllos Afouras, and Kristen Grauman. Video-mined task graphs for keystone recognition in instructional videos. *Advances in Neural Information Processing Systems*, 36, 2024. 7
- [5] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 105–121, 2018. 2
- [6] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 5
- [7] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 847–859, 2021. 1
- [8] Yizhak Ben-Shabat, Jonathan Paul, Eviatar Segev, Oren Shtrout, and Stephen Gould. Ikea ego 3d dataset: Understanding furniture assembly actions from ego-view 3d point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4355–4364, 2024. 1
- [9] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 6, 7
- [10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 4, 5
- [11] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 2
- [12] Chien-Yi Chang, De-An Huang, Danfei Xu, Ehsan Adeli, Li Fei-Fei, and Juan Carlos Niebles. Procedure planning in instructional videos. In *European Conference on Computer Vision*, pages 334–350. Springer, 2020. 2
- [13] Dibiyadip Chatterjee, Fadime Sener, Shugao Ma, and Angela Yao. Opening the vocabulary of egocentric actions. *Advances in Neural Information Processing Systems*, 36: 33174–33187, 2023. 2
- [14] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418, 2024. 1, 2, 6, 7, 8
- [15] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2
- [16] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, pages 1–23, 2022. 6
- [17] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part V 14*, pages 269–284. Springer, 2016. 1, 2
- [18] Guodong Ding, Fadime Sener, Shugao Ma, and Angela Yao. Every mistake counts in assembly. *arXiv preprint arXiv:2307.16453*, 2023. 2
- [19] Guodong Ding, Fadime Sener, and Angela Yao. Temporal action segmentation: An analysis of modern techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1, 2
- [20] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 2, 3, 6, 7
- [21] Nikita Dvornik, Isma Hadji, Hai Pham, Dhairat Bhatt, Brais Martinez, Afsaneh Fazly, and Allan D Jepson. Flow graph to video grounding for weakly-supervised multi-step localization. In *European Conference on Computer Vision*, pages 319–335. Springer, 2022. 3
- [22] Alessandro Flaborea, Guido Maria D’Amely di Melendugno, Leonardo Plini, Luca Scofano, Edoardo De Matteis, Antonino Furnari, Giovanni Maria Farinella, and Fabio Galasso. Prego: online mistake detection in procedural egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18483–18492, 2024. 2
- [23] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 6252–6261, 2019. 2

- [24] Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018. 6
- [25] Mingfei Gao, Mingze Xu, Larry S Davis, Richard Socher, and Caiming Xiong. Startnet: Online detection of action start in untrimmed videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5542–5551, 2019. 2
- [26] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13505–13515, 2021. 2
- [27] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2
- [28] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024. 2, 3, 4, 6
- [29] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. Ma-lmm: Memory-augmented large multimodal model for long-term video understanding. *CVPR*, 2024. 1, 2, 3
- [30] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2
- [31] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekeshe, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What’s the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024. 1, 3
- [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. In *International Conference on Learning Representations*, 2022. 6
- [33] Md Mohaiminul Islam, Tushar Nagarajan, Huiyu Wang, Fu-Jen Chu, Kris Kitani, Gedas Bertasius, and Xitong Yang. Propose, assess, search: Harnessing llms for goal-oriented planning in instructional videos. *arXiv preprint arXiv:2409.20557*, 2024. 2
- [34] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. <http://csrcv.ucf.edu/THUMOS14/>, 2014. 1
- [35] Mahnaz Koupaee and William Yang Wang. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*, 2018. 3
- [36] Bolin Lai, Sam Toyer, Tushar Nagarajan, Rohit Girdhar, Shengxin Zha, James M Rehg, Kris Kitani, Kristen Grauman, Ruta Desai, and Miao Liu. Human action anticipation: A survey. *arXiv preprint arXiv:2410.14045*, 2024. 1
- [37] Shih-Po Lee, Zijia Lu, Zekun Zhang, Minh Hoai, and Ehsan Elhamifar. Error detection in egocentric procedural task videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18655–18666, 2024. 2
- [38] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2, 3, 7
- [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. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 2, 4, 8
- [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 preprint arXiv:2305.06355*, 2023. 2
- [41] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 2, 7
- [42] Han Lin, Tushar Nagarajan, Nicolas Ballas, Mido Assran, Mojtaba Komeili, Mohit Bansal, and Koustuv Sinha. Vedit: Latent prediction architecture for procedural video representation learning. *arXiv preprint arXiv:2410.03478*, 2024. 6, 7
- [43] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13853–13863, 2022. 2, 7
- [44] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2023. 2, 3
- [45] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 2, 8
- [46] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12: 157–173, 2024. 1, 3
- [47] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 2
- [48] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of*

- the *IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019. 2
- [49] Medhini Narasimhan, Arsha Nagrani, Chen Sun, Michael Rubinstein, Trevor Darrell, Anna Rohrbach, and Cordelia Schmid. Tl; dw? summarizing instructional videos with task relevance and cross-modal saliency. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. 2
- [50] Medhini Narasimhan, Licheng Yu, Sean Bell, Ning Zhang, and Trevor Darrell. Learning and verification of task structure in instructional videos. *arXiv preprint arXiv:2303.13519*, 2023. 7
- [51] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 7
- [52] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 2, 4
- [53] Zhanzhong Pang, Fadime Sener, and Angela Yao. Context-enhanced memory-refined transformer for online action detection. *arXiv preprint arXiv:2503.18359*, 2025. 1, 2
- [54] Dhruv Patel, Hamid Eghbalzadeh, Nitin Kamra, Michael Louis Iuzzolino, Unnat Jain, and Ruta Desai. Pre-trained language models as visual planners for human assistance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15302–15314, 2023. 3
- [55] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivan Agrawal, and Jeff Dean. Efficiently scaling transformer inference. *Proceedings of Machine Learning and Systems*, 5:606–624, 2023. 5
- [56] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5285–5297, 2023. 7
- [57] Alec Radford. Improving language understanding by generative pre-training. *Openai*, 2018. 5
- [58] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 2
- [59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3, 4
- [60] Francesco Ragusa, Antonino Furnari, Salvatore Livatino, and Giovanni Maria Farinella. The meccano dataset: Understanding human-object interactions from egocentric videos in an industrial-like domain. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1569–1578, 2021. 1
- [61] Abhinav Rai, Fadime Sener, and Angela Yao. Transformed rois for capturing visual transformations in videos. *Computer Vision and Image Understanding*, 224:103558, 2022. 2
- [62] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14313–14323, 2024. 2
- [63] Hamid Rezaatofghi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019. 5
- [64] Luigi Seminara, Giovanni Maria Farinella, and Antonino Furnari. Differentiable task graph learning: Procedural activity representation and online mistake detection from egocentric videos. In *Advances in Neural Information Processing Systems*, pages 59373–59407. Curran Associates, Inc., 2024. 3
- [65] Fadime Sener and Angela Yao. Zero-shot anticipation for instructional activities. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 862–871, 2019. 1, 2
- [66] Fadime Sener, Dipika Singhanian, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 154–171. Springer, 2020. 2, 6
- [67] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhanian, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022. 1, 2, 3, 6
- [68] Md Salman Shamil, Dibyadip Chatterjee, Fadime Sener, Shugao Ma, and Angela Yao. On the utility of 3d hand poses for action recognition. In *European Conference on Computer Vision*, pages 436–454. Springer, 2024. 4
- [69] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9869–9878, 2020. 4, 5, 6
- [70] Zheng Shou, Junting Pan, Jonathan Chan, Kazuyuki Miyazawa, Hassan Mansour, Anthony Vetro, Xavier Giro-i Nieto, and Shih-Fu Chang. Online detection of action start in untrimmed, streaming videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 534–551, 2018. 1, 2
- [71] Enxin Song, Wenhao Chai, Guan hong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024. 2

- [72] Yale Song, Eugene Byrne, Tushar Nagarajan, Huiyu Wang, Miguel Martin, and Lorenzo Torresani. Ego4d goal-step: Toward hierarchical understanding of procedural activities. *Advances in Neural Information Processing Systems*, 36, 2023. 2, 3, 4, 6
- [73] Reuben Tan, Ximeng Sun, Ping Hu, Jui-hsien Wang, Hanieh Deilamsalehy, Bryan A Plummer, Bryan Russell, and Kate Saenko. Koala: Key frame-conditioned long video-llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13581–13591, 2024. 2
- [74] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 2, 6
- [75] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 3
- [76] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9568–9578, 2024. 3
- [77] 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 preprint arXiv:2302.13971*, 2023. 2
- [78] Huiyu Wang, Mitesh Kumar Singh, and Lorenzo Torresani. Ego-only: Egocentric action detection without exocentric transferring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5250–5261, 2023. 6
- [79] Jiahao Wang, Guo Chen, Yifei Huang, Limin Wang, and Tong Lu. Memory-and-anticipation transformer for online action understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13824–13835, 2023. 1, 2, 3, 4, 6
- [80] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. *arXiv preprint arXiv:2403.10517*, 2024. 2
- [81] Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. Longvlm: Efficient long video understanding via large language models. *arXiv preprint arXiv:2404.03384*, 2024. 2, 5
- [82] Wikihow. How to do anything. <http://www.wikihow.com/>, 2005. 1
- [83] Shiwei Wu, Joya Chen, Kevin Qinghong Lin, Qimeng Wang, Yan Gao, Qianli Xu, Tong Xu, Yao Hu, Enhong Chen, and Mike Zheng Shou. Videollm-mod: Efficient video-language streaming with mixture-of-depths vision computation. *arXiv preprint arXiv:2408.16730*, 2024. 1, 2, 6, 7, 8
- [84] Junbin Xiao, Angela Yao, Yicong Li, and Tat-Seng Chua. Can i trust your answer? visually grounded video question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13204–13214, 2024. 3
- [85] Junbin Xiao, Nanxin Huang, Hangyu Qin, Dongyang Li, Yicong Li, Fengbin Zhu, Zhulin Tao, Jianxing Yu, Liang Lin, Tat-Seng Chua, et al. Videoqa in the era of llms: An empirical study. *International Journal of Computer Vision*, pages 1–24, 2025. 3
- [86] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. *Advances in Neural Information Processing Systems*, 34:1086–1099, 2021. 1, 2, 3, 4, 6
- [87] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 3
- [88] Olga Zatsarynna and Juergen Gall. Action anticipation with goal consistency. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 1630–1634. IEEE, 2023. 6
- [89] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 2, 3, 4
- [90] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Is an object-centric video representation beneficial for transfer? In *Proceedings of the Asian Conference on Computer Vision*, pages 1976–1994, 2022. 2
- [91] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Helping hands: An object-aware ego-centric video recognition model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13901–13912, 2023. 4
- [92] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 543–553, 2023. 2
- [93] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkan Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024. 3
- [94] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 2
- [95] He Zhao, Isma Hadji, Nikita Dvornik, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. P3iv: Probabilistic procedure planning from instructional videos with weak supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2938–2948, 2022. 2
- [96] Qi Zhao, Shijie Wang, Ce Zhang, Changcheng Fu, Minh Quan Do, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. AntGPT: Can large language models help long-term action anticipation from videos? In *The Twelfth International Conference on Learning Representations*, 2024. 2

- [97] Yue Zhao and Philipp Krähenbühl. Real-time online video detection with temporal smoothing transformers. In *European Conference on Computer Vision*, pages 485–502. Springer, 2022. [1](#), [2](#), [3](#), [6](#)
- [98] Yue Zhao, Long Zhao, Xingyi Zhou, Jialin Wu, Chun-Te Chu, Hui Miao, Florian Schroff, Hartwig Adam, Ting Liu, Boqing Gong, et al. Distilling vision-language models on millions of videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13106–13116, 2024. [2](#)
- [99] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14825–14835, 2023. [7](#)
- [100] Honglu Zhou, Roberto Martín-Martín, Mubbasir Kapadia, Silvio Savarese, and Juan Carlos Niebles. Procedure-aware pretraining for instructional video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10727–10738, 2023. [1](#), [2](#), [3](#), [7](#)
- [101] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. [1](#), [2](#)
- [102] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. [2](#)
- [103] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3537–3545, 2019. [2](#)