This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Open-ended Hierarchical Streaming Video Understanding with Vision Language Models

Hyolim Kang* Yunsu Park* Youngbeom Yoo Yeeun Choi Seon Joo Kim
Yonsei University
{hyolimkang,ysp7954,youngbeom.yoo,yeeun-choi,seonjookim}@yonsei.ac.kr

Abstract

We introduce Hierarchical Streaming Video Understanding, a task that combines online temporal action localization with free-form description generation. Given the scarcity of datasets with hierarchical and fine-grained temporal annotations, we demonstrate that LLMs can effectively group atomic actions into higher-level events, enriching existing datasets. We then propose OpenHOUSE (Open-ended Hierarchical Online Understanding System for Events), which extends streaming action perception beyond action classification. OpenHOUSE features a specialized streaming module that accurately detects boundaries between closely adjacent actions, nearly doubling the performance of direct extensions of existing methods. We envision the future of streaming action perception in the integration of powerful generative models, with OpenHOUSE representing a key step in that direction.

1. Introduction

Imagine a cyclist wearing an egocentric camera while repairing a bicycle. If we are designing a system to assist the cyclist, it must not only recognize the fine-grained action of "tightening a bolt," but also understand that it is a substep of "changing the wheel," which in turn contributes to the larger goal of "repairing the bike." Recognizing the importance of the hierarchical structure, we argue that such understanding must be acquired instantaneously for effective real-time assistance. Given this complexity, relying on predefined class categories is suboptimal, as their limited expressiveness restricts the system's ability to capture the hierarchical structure in detail.

This highlights the need for an instantaneous, open-ended and hierarchical understanding of streaming input (Figure 1). Large-scale Vision Language Models (VLMs) [1, 27, 28], with their powerful reasoning capabilities, are well-suited for handling this challenge. However, applying VLMs to continuous visual streams presents unique challenges beyond standard offline video understanding [25, 29]. First, the model must determine "when" to generate output within the task hierarchy. Continuously calling the VLM is computationally expensive, and generating descriptions for every frame would result in excessive, fragmented, and inaccurate descriptions. Second, given the untrimmed, extensive nature of the video, the model must determine which segment is relevant to the ongoing action, as processing the entire history is impractical. Finally, using this information, the model should provide an immediate description of the ongoing action.

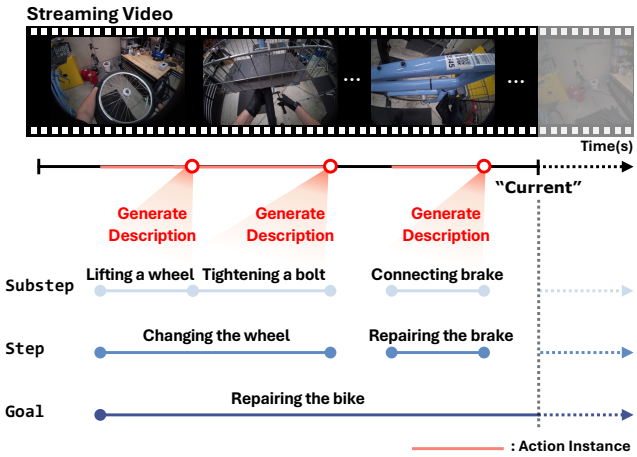


Figure 1. Illustration of the Hierarchical Streaming Video Understanding task. When action instances are detected, the corresponding hierarchical descriptions are generated instantly. The hierarchy consists of three levels: *Goal*, *Step*, and *Substep*. The *Goal* description can be generated before the entire video concludes.

Rather than devising a single model, we decouple the task into two parts: (i) a lightweight "streaming" perception module that processes each frame and performs hierarchy-aware action detection in real time, calling the VLM only when needed, and (ii) a VLM that generates descriptions only when called by the perception module. This disentanglement offers distinct advantages over a single model approach. First, it enables efficient operation by continuously running the lightweight module and sparingly acti-

*equal contribution

vating the computationally expensive VLM. Second, separating streaming perception tasks from caption generation allows the VLM to function as originally trained, enabling zero-shot inference. By avoiding fine-tuning or task-specific adaptation, we preserve the VLM's generalization ability and avoid the degradation typically caused by tuning on relatively small custom datasets.

The proposed system, **Open**-ended **H**ierarchical **O**nline **U**nderstanding **S**ystem for **E**vents (**OpenHOUSE**), is an approach that bridges the gap between established online action detection methods and large-scale VLMs. Without compromising the strict online nature of existing online action detection formulation, we effectively incorporate the generative capabilities of large-scale VLMs by employing a lightweight online model as a *VLM caller* and an *online hierarchical action segmentor* to determine both which parts of the streaming video should be processed and their hierarchical structure. This design distinguishes our approach from previous methods that adopted VLMs for streaming perception, such as relying on fixed decoding points [51] or invoking the VLM for every incoming frame [5].

However, a straightforward integration of existing online modules with large-scale VLMs proves insufficient due to the following challenges:

Scarcity of hierarchical annotations: While VLMs do not require fine-tuning to grasp hierarchical structures, detailed temporal annotations that include hierarchical relationships are essential for training a streaming module to understand action hierarchies. However, although many datasets [3, 8, 13, 31] offer atomic action annotations, hierarchical annotations remain scarce. To address this, we propose a dataset generation pipeline that leverages the reasoning capacity of LLMs to group atomic actions into higher-level hierarchical structures. Notably, experiments on Ego4D-GoalStep [34], which has ground-truth hierarchical annotations, show that models trained on the generated dataset perform competitively with those trained on ground truth, confirming the reliability of our dataset generation pipeline. By training the streaming module with these annotations, our dataset pipeline effectively distills LLM's hierarchical awareness to the streaming module.

Class-agnostic action boundary modeling: Similar to class-agnostic object detectors in open-vocabulary recognition [7, 12, 36], the streaming module in OpenHOUSE operates without predefined class labels. However, previous class-agnostic action boundary detection strategy—based on detecting transitions from background to action frames—significantly struggles in procedural videos due to the lack of sufficient background frames between actions. To address this, we propose a new strategy that combines estimated action presence (or "actionness") with progression to identify action start and end points. This approach has proven highly effective, nearly doubling the F1 score compared to conventional actionness-based methods [2, 50] in the OpenHOUSE framework.

Our contributions can be summarized as follows:

- We introduce the new, practical problem of Open-ended Hierarchical Streaming Video Understanding and propose **OpenHOUSE** as a solution.
- We validate the capability of LLMs to group atomic actions into higher-level annotations, addressing the scarcity of hierarchical temporal data.
- We propose a novel strategy for detecting action boundaries, capable of handling instances that are strictly adjacent without background frames.
- We conduct extensive experiments including the cross-dataset evaluation, which demonstrates the generalizability and scalability of our approach.

2. Related Work

2.1. Streaming Action Perception

Online Action Detection (OAD) [9] is a well-established task in streaming video understanding that involves identifying the action class of the current frame in a streaming setting. While extensive research has focused on OAD [2, 4, 10, 40, 43, 44, 46, 49, 50], its frame-centric approach falls short for real-world streaming video analysis, which requires event-level recognition.

Conversely, Online Temporal Action Localization (On-TAL) [18–20, 30, 35, 37], the task recently introduced in [17], targets real-time event detection. Beyond the constraint of unobserved future frames, On-TAL requires immediate responsiveness, producing detection results as soon as an event terminates. This unique requirement makes the problem more challenging, preventing the naive extension of previous approaches to On-TAL.

Our Work Targeting practical online scenarios, we adopt the OAD-based On-TAL setting [17]. While a relevant recent study [49] explores VLMs for per-frame action classification, we focus on leveraging VLMs' free-form generation and reasoning capabilities to capture the hierarchical structure of ongoing actions.

2.2. Streaming Open-ended Video Understanding

Video Captioning Many works have focused on generating captions that describe video content. Notably, SDVC [51] adopted the concept of a "decoding point," which enables captions to be generated without observing the full video. However, unlike OAD or On-TAL, SDVC is not designed for prompt responses in online scenarios. Its primary innovation lies in enabling caption generation without full video access, rather than recognizing ongoing actions and responding in real time.

Video Question Answering (VQA) The application of VLMs for VQA [25] has gained increasing attention due



Figure 2. In the SDVC setting, action instances are generated at fixed decoding points (e.g., D3), which occur at regular intervals, inevitably introducing a delay (D3 - T2). In contrast, the On-TAL formulation requires action instance detection at T2, ensuring real-time responsiveness. A systematic analysis of this delay is provided in Sec. 4.2.

to their strong reasoning abilities. While many studies address conventional, "offline" VQA, the streaming VQA scenario—despite its practical importance—has been relatively underexplored. Recent works [5, 41] have enabled streaming VQA by fine-tuning VLMs with timestamp-aligned custom VQA datasets. However, these frameworks require the fine-tuned VLM to perform inference on *every incoming frame* and to handle all temporal recognition tasks and text generation by one model.

Our Work Following the On-TAL problem definition [17], our approach runs in a strict online setting: open-ended descriptions must be generated instantly in response to the current action status, whereas in [51], decoding points are not required to align with ongoing actions (Figure 2). Furthermore, by decoupling temporal understanding from text generation, OpenHOUSE achieves a more sophisticated, hierarchical temporal comprehension, enabling it to capture structured *action intervals* rather than aligning predictions to single points in time [5, 41].

2.3. Hierarchical Video Understanding

Hierarchical video understanding seeks to recognize actions across multiple levels of granularity, from fine-grained actions to higher-level activities. Although this hierarchical structure aligns with natural action dynamics, few recent works address this topic [15, 34].

Our Work To the best of our knowledge, OpenHOUSE is the first framework to explore hierarchical understanding in a streaming setting. Unlike [15], which focuses on generating high-quality captions for a given clip or segment by recurrently leveraging low-level captions, our approach emphasizes identifying the "temporal boundaries" of action instances across different hierarchical levels.

3. Target Task and Our Approach

3.1. Hierarchical Streaming Video Understanding

Consider an untrimmed video $V = \{x_\tau\}_{\tau=1}^{T}$ with M hierarchical action instances $\Psi = \{\psi_m\}_{m=1}^{M} = \{(t_m^s, t_m^e, d_m, h_m)\}_{m=1}^{M}$ provided in a streaming format. Here, x_τ denotes the τth frame; $t_m^s \in \mathbb{R}$, $t_m^e \in \mathbb{R}$ are the start and the end timestep; and d_m and $h_m \in \mathbb{N}$ represent the (possibly free-form) description and hierarchical level of the mth action instance ψ_m respectively. To

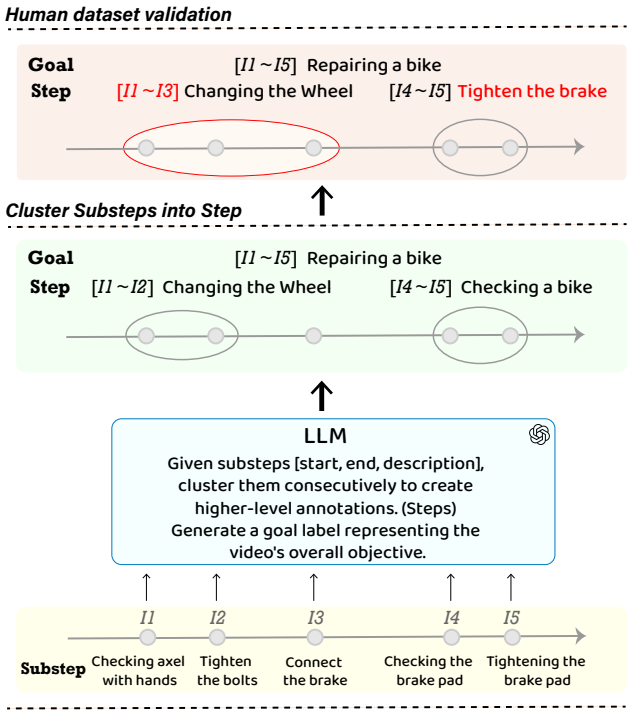


Figure 3. Dataset generation pipeline. Given substep instances from $I1$ to $I5$, the LLM clusters these substeps into three steps, generating step descriptions, timestamps, and a goal description. Afterward, human validation is conducted to include the missing annotation $I3$ into the step annotations of $I1$ to $I2$, and to revise the step description for $I4$ to $I5$.

ensure immediate action recognition, the system must respond promptly to action termination. We therefore adopt the OAD based On-TAL formulation [17, 18, 37] for this task. Note that recent TAL-based On-TAL formulations [19, 30, 35] relax online constraints through anchor-based boundary regression, allowing some flexibility in the timing of action instance generation, which can result in either early or delayed predictions. In contrast, the OAD-based formulation adheres to a per-frame grouping paradigm that strictly aligns action termination with the current frame, which is why we opted for this setting.

Our objective is to generate and accumulate action proposals ψ as soon as each action completes, reconstructing Ψ *without retrospective modification* to any ψ. This work focuses on a three-level hierarchy: $substep, step, goal$, corresponding to $h = 1, 2, 3$ respectively. $substep$ and $step$ are defined at the instance level with temporal annotations, while the $goal$ applies to the entire video.

3.2. Dataset Pipeline

To enable the streaming module to recognize hierarchical structures, both hierarchical action annotations and fine-grained temporal annotations are essential. While some datasets offer either fine-grained atomic action annotations

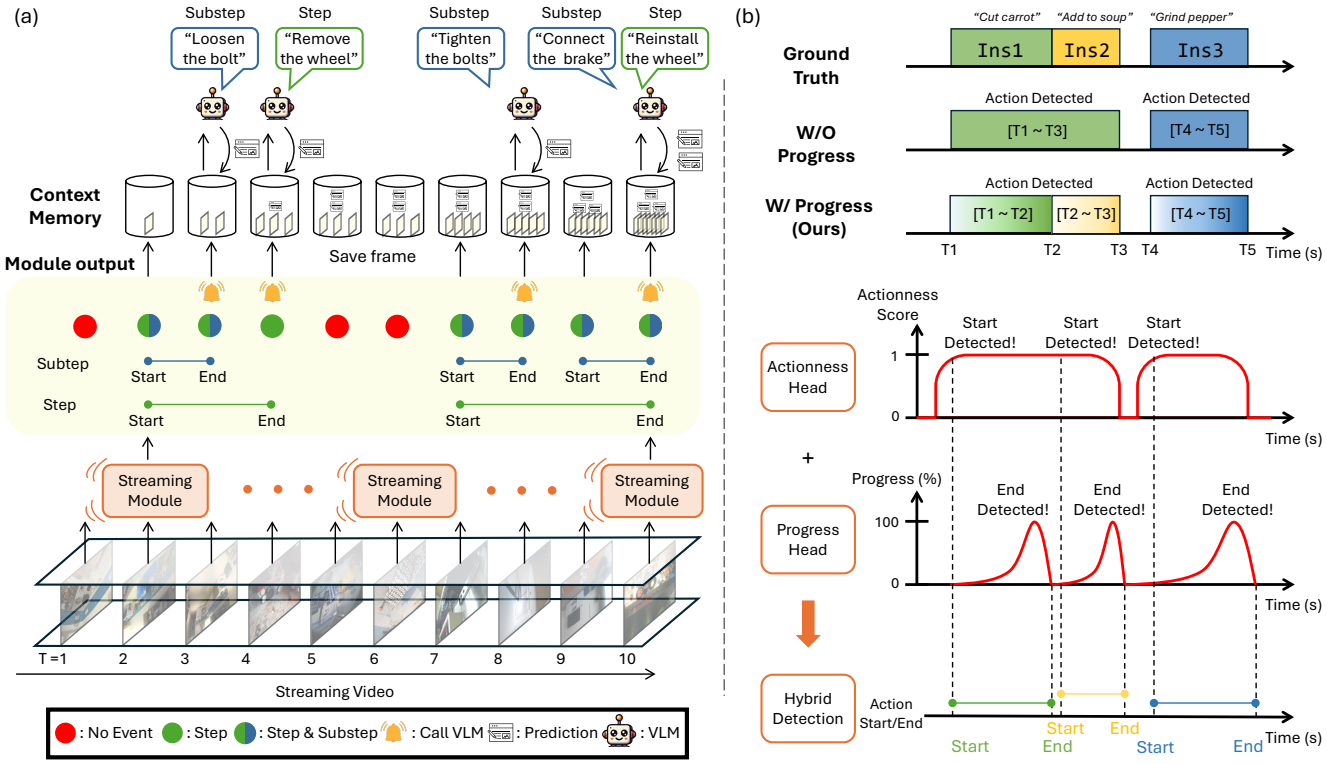


Figure 4. (a) Illustration of OpenHOUSE. The lightweight Streaming module processes every frame, while VLM inference is triggered selectively at $T = 3, 4, 8, 10$, indicated by the yellow bell icon. (b) Diagram depicting how our hybrid action boundary detection works. For instance, at $T2$, a sudden drop in the progression signals the action end, marking the current frame as a background frame. In the next timestep, a high actionness score ($= 1.0$) classifies the frame as an action frame, resulting in a background-to-action transition and marking it as the start of `Ins2`.

[8, 13] or hierarchical captions [15], very few provide both. To the best of our knowledge, the Ego-4D Goalstep [34] dataset is the only hierarchical action recognition dataset with sufficiently fine-grained temporal annotations. To address this limitation, we leverage the reasoning capacity of LLMs. Just as humans can intuitively group atomic action annotations into higher-level categories, we utilize an LLM to perform this hierarchical grouping automatically.

Figure 3 illustrates the overall framework of our dataset generation pipeline, which consists of two main stages: clustering substeps and human validation. In the clustering substeps stage, we employ an LLM to group atomic actions, or *substeps*, into meaningful clusters, referred to as *steps*. After this initial clustering, a post-processing step is applied to further refine the results. Subsequently, the LLM generates appropriate summaries for each set of substeps to form the step-level annotations, and also produces goal-level annotations that best capture the overall purpose of the video. For more organized descriptions, we applied K-means clustering to group similar steps (200 for Ego-Exo4D Keystep [13] and 500 for Epic-Kitchens 100 [8]). The LLM then generated representative captions for each group to categorize the step annotations.

In the human validation stage, generated annotations are evaluated and adjusted across two aspects: temporal consistency and caption quality. Temporal consistency involves checking missing instances, as well as addressing abnormally long or short temporal annotations. For caption quality, human evaluators revise step-level captions that fail to adequately reflect the content of their substeps, fill in missing captions, and adjust goal captions when they do not accurately convey the video's overall purpose. Empirically, we found that *minimal* adjustment is needed in this stage due to the powerful reasoning capabilities of LLMs.

Using this pipeline, we successfully augmented the Ego-Exo4D Keystep [13] dataset and the Epic-Kitchens 100 [8] dataset into hierarchical formats, thereby addressing the scarcity of hierarchical annotations. In Section 4, we further demonstrate the validity of our generated annotations. These annotations will be made public along with the code.

3.3. Approach: OpenHOUSE

Overview Figure 4 (a) illustrates the schematic diagram of **Open**-ended **H**ierarchical **O**nline **U**nderstanding **S**ystem for **E**vents (**OpenHOUSE**), which consists of three main components: the Streaming module, Context memory, and

the frozen VLM. The Streaming module processes each incoming frame, performing class-agnostic On-TAL based on OAD [18]. As an extended OAD model, this module also tracks ongoing action instances at different hierarchical levels, determining whether to store the incoming frame in the context memory.

When the module detects the termination of an action instance, it returns a class-agnostic action instance along with its hierarchy level. This action instance serves as a "query" to retrieve relevant information from the context memory, such as frames and previous predictions. The retrieved information, consisting of relevant frames and previous text-based predictions, is then passed to the frozen VLM, along with prompts tailored to the action instance's hierarchy level, to generate an open-ended description of the segment. The generated prediction is subsequently added to the context memory for future reference. The key feature is that the lightweight Streaming module processes every frame, while the computationally heavy VLM is used sparingly—resulting in an efficient setup for online scenarios.

3.3.1. Streaming Module

Motivation The Streaming module in OpenHOUSE performs class-agnostic On-TAL, but faces unique challenges in identifying step and substep boundaries. Prior approaches for class-agnostic On-TAL [17, 18] detect action instances by grouping action frames—a straightforward method when action instances are sparsely distributed with sufficient background frames naturally separating them (`Ins2` and `Ins3` in Figure 4 (b)). By simply detecting transitions between background frame and action frames, these methods can identify action boundaries effectively.

Nevertheless, in procedural videos, which are the primary focus of our work, action instances often occur in close succession with minimal or no background frames in between (`Ins1` and `Ins2` in Figure 4 (b)). Following prior methods in this context leads to a monotonous sequence of action frames, interpreting multiple instances as a single continuous instance (`w/o Progress` in Figure 4 (b)). Even when few background frames exist between action instances, distinguishing them is still challenging. Ideally, both action ends and starts should be marked as action frames, with only the short background marked as background—a distinction that is theoretically possible but extremely difficult in practice. This issue is unique to the class-agnostic setting; in class-aware settings, boundaries can be clearly marked by changes in action labels [32], but such cues are absent in the class-agnostic setting.

Hybrid action boundary detection To address this issue, we introduce a novel strategy for determining action boundaries. In addition to measuring actionness, we also measure the "progress" of action instances and detect sudden drops in this progress to signify action ends. Note that ground-truth progress can be derived from temporal annotations (e.g., if an action occurs between 2s and 4s, the timestamp at 3s corresponds to 50% progress), and it is used for training progression heads. Unlike prior methods that use progress as auxiliary information [32], we explicitly leverage the estimated progress score for action boundary detection.

However, using progress alone for action boundary detection is not feasible, as it increases gradually at the start, providing no distinctive signal for detecting action start. Thus, we adopt a hybrid approach: action starts are detected using conventional actionness-based methods, while action ends are identified by monitoring progress drops.

Figure 4 (b) illustrates our action boundary detection algorithm within a single hierarchy. Action starts and ends are detected separately by the actionness head and progress head, respectively. For multiple hierarchies, this approach can be extended by assigning additional progress head to each hierarchy and incorporating a state-emitting head [18] instead of binary actionness head.

Implementation Details Following recent OAD literature [2], we use a simple RNN backbone shared across three heads: a state-emitting head, a progression head for steps, and a progression head for substeps. The state-emitting head is trained using standard cross-entropy loss, as in [18]. Although the progression heads are essentially regression tasks, we apply histogram loss [11, 14] to transform them into classification problems, allowing all heads to be trained with cross-entropy loss.

3.3.2. VLM

Exploiting the fact that our Streaming module sparingly generates action instances, we use each instance generation as a signal to call the VLM, enabling occasional VLM inference. In this setup, the VLM is solely responsible for generating captions based on provided frames and text, with no special online behavior required. This allows us to leverage the VLM's strong zero-shot inference capability, which enables the VLM to remain frozen throughout training. Additionally, inspired by RAG [21], we incorporate information from previous predictions instead of relying solely on the current action instance, assuming a context memory that stores relevant past frames and predictions.

For example, assume that the Streaming module detects the end of a *step*. Then, the module will provide the action start and current timestamps along with the corresponding hierarchy level: *step*. Using these, relevant data—such as previous *step* predictions and frames of detected *substeps* within that interval—is retrieved from the context memory. Substep frames are sampled from these relevant *substeps* instead of using uniform sampling, which could overlook important details. This, along with a customized prompt for the *step* hierarchy, is passed to the VLM, producing appropriate description. The prediction is then added to the context memory for future use. Exact prompts and other details are provided in the supplementary materials.

4. Experiment and Analysis

4.1. Experimental Setup

Datasets We begin our evaluation with Ego4D-GoalStep (EgoGS) [34], which includes ground truth hierarchical temporal annotations. After validating our dataset generation pipeline, we extend our experiments to Ego-Exo4D Keystep (EgEx) [13] and Epic-Kitchens 100 (EK100) [8]. Detailed dataset analyses, including statistics, are provided in the supplementary materials. Unless specified, hereafter "EgEx" refers to Ego-view in Ego-Exo4D Keystep dataset.

Features Following established conventions in video temporal understanding [9, 22, 30, 35, 48], we use a pretrained vision encoder to extract features from raw frames. Except for the exo view of EgoExo4D, which uses the official features, we use the Egovideo [26] encoder. All experiments, except for the exo view of EgoExo4D, are carried out with the *same features* to ensure fair comparison. Note that features are extracted online, and we take this into account when measuring inference speed.

Evaluation Metric Our evaluation protocol focuses on three aspects: (i) class-agnostic localization ability of the Streaming module, (ii) quality and alignment of generated descriptions, and (iii) joint assessment of both criteria.

Based on the problem formulation (Section 3.1), our method is an open-ended extension of OAD based On-TAL. Instead of simply predicting class indices, it evolves to generate open-ended descriptions of the target action instance. Thus, following a recent work [18, 45, 51], we opt for F1@$tIOU$ as our main metric to evaluate the class-agnostic localization performance of the Streaming module.

For description quality and alignment, we use GPT-Score, a standard GPT-assisted evaluation for open-ended VLM outputs [16, 23, 24, 33, 42]. Following [25], we score all true positive predictions on a 1–5 scale across multiple criteria and report the average GPT-Score in experiments. As noted in [52], we also found that conventional N-gram metrics (e.g., CIDEr [38]) are not the optimal choice for zero-shot VLM evaluation.

To jointly evaluate localization ability and description alignment, we adopt a zero-shot evaluation strategy from CLIP [27], encoding the predicted and all ground truth descriptions into vectors with the GPT-4 text encoder [1] and ranking them by pairwise cosine similarity; a prediction is a true positive only if it exceeds the $tIOU$ threshold AND its corresponding ground truth description appears in the top-k matches. Hereafter, we refer to the F1 metric for class-agnostic localization as F1 (loc.), and top-k F1 metric as F1 (loc.+desc.). Detailed evaluation steps, full GPT-Score results without averaging, and failure cases of N-gram-based metrics are provided in the supplementary materials.

	F1 (loc.) ↑			F1 (loc. + desc.) ↑			GPT Score ↑
	0.3	0.5	0.7	0.3	0.5	0.7	
Annotation							
EgoGs Pseudo	64.7	56.1	42.8	34.04	30.87	26.10	2.98
Train							
EgoGS GT	51.58	39.70	24.76	15.23	12.67	8.68	2.96
EgoGS Pseudo	50.65	38.24	23.48	14.30	11.63	7.66	3.02

Table 1. Evaluation of pseudo-labels from our dataset pipeline. All F1 scores are calculated using the ground truth step labels from the EgoGS dataset.

Dataset	Annotation	Hier.	Hybrid	F1 (loc.) ↑
EgoGS	GT	Step	✗	16.78
			✔	**39.35**
		Substep	✗	22.48
			✔	**44.79**
	Pseudo	Step	✗	25.34
			✔	**37.65**
EgEx	GT	Substep	✗	8.42
			✔	**52.2**
	Pseudo	Step	✗	6.12
			✔	**46.33**
EK100	GT	Substep	✗	31.46
			✔	**48.95**
	Pseudo	Step	✗	19.91
			✔	**34.17**

Table 2. Ablation study of Hybrid Action Boundary Detection. Since EgEx and EK100 lack ground truth step labels, experiments are conducted using pseudo-step labels.

4.2. Results and Analysis

Validating Dataset Pipeline Table 1 presents the validation of our dataset pipeline. Since the Ego4D-Goalstep dataset contains both substep and step annotations, we compare the generated pseudo step labels from our pipeline with the ground truth step labels. The results in the first row show that the LLM effectively groups atomic actions, leading to a reasonable approximation of the original step annotations. However, due to the inherent subjectivity in action temporal boundaries, perfect recovery is obviously impossible.

More importantly, we assess whether these pseudo-labels provide *meaningful learning signals* for step-level hierarchy training. Surprisingly, models trained with our pseudo-labels performed competitively with those trained on the actual ground truth labels. This suggests that even though the generated step labels do not exactly match the originals, they do not contain critical errors but instead provide a plausible alternative. Just as different human annotators might produce varying temporal boundaries, the generated labels still offer a valid interpretation rather than a completely incorrect one.

Hybrid Action Boundary Detection Table 2 demonstrates the effectiveness of our hybrid action boundary detection method. Across all configurations and datasets, incorporating our hybrid method yields significant gains. It highlights the critical importance of our approach in processing proce-

Method	Train	Valid	Hier.	F1 (loc.) ↑			F1 (loc. + desc.) ↑			GPT Score ↑	Goal Acc. ↑
				0.3	0.5	0.7	0.3	0.5	0.7		
GT Proposal	-	EgoGS	Step	-	-	-	28.68	28.68	28.68	2.96	47.01
			Substep	-	-	-	33.12	33.12	33.12	2.77	
TeSTra [50]	EgoGS	EgoGS	Step	18.27	12.51	7.08	4.21	3.12	2.19	2.79	37.31
			Substep	22.56	15.85	9.55	6.84	4.81	2.72	2.74	
MiniROAD [2]			Step	19.23	12.46	6.56	4.26	2.81	1.79	2.73	38.31
			Substep	32.12	21.90	12.88	9.77	7.05	4.14	2.77	
MAT [39]			Step	18.36	12.34	7.00	4.01	3.01	1.98	2.82	40.30
			Substep	26.22	18.91	11.28	7.32	5.13	3.21	2.74	
SDVC [51]			Step	22.23	10.43	3.06	2.61	1.32	0.57	2.12	-
			Substep	17.81	7.79	2.30	2.28	0.80	0.22	1.76	
SDVC* [51]			Step	22.23	10.43	3.06	6.20	3.25	0.79	2.45	-
			Substep	17.81	7.79	2.30	5.73	2.57	0.93	2.18	
OpenHOUSE	EgoGS		Step	**51.58**	**39.70**	**24.76**	**15.23**	**12.67**	**8.68**	2.96	**47.76**
			Substep	55.17	43.71	28.83	19.79	16.11	10.89	2.78	
	EgoGS†		Step	50.65	38.24	23.48	14.30	11.63	7.66	3.02	45.52
			Substep	**55.75**	**43.88**	**29.38**	**19.80**	**16.42**	**11.71**	2.79	
OpenHOUSE	EgEx†	EgEx†	Step	62.85	45.75	26.15	19.99	14.76	9.87	3.18	90.66
			Substep	63.42	51.12	34.36	23.22	19.53	14.03	2.56	
	EK100†	EK100†	Step	55.32	34.48	15.91	14.13	9.04	4.35	3.24	29.71
			Substep	65.98	48.01	25.73	13.54	10.51	6.09	2.64	
	Exo.†	Exo.†	Step	62.03	39.73	19.73	12.91	9.18	4.99	2.80	79.12
			Substep	57.33	40.29	21.35	10.06	7.60	4.50	1.96	

Table 3. Comparison with baseline methods and results in various datasets. † indicates datasets with "step pseudo-label" from our pipeline, and "Exo." denotes EgEx's Exo view. "GT proposal" refers to experiments using ground truth action proposals as VLM input, without class information. Zero-shot VLM inference achieves around 30% accuracy in matching ground truth labels. InternVL2-40B-AWQ [6] is used as the frozen VLM. Following the original implementation, YT-Temporal [47] pretrained Vid2Seq [45] weights are used for initialization of SDVC, and fine-tuned using EgoGS annotations, with separate training for each hierarchy. We used the default configuration on SDVC.

Train	Valid	Hier.	F1 (loc.) ↑			F1 (loc. + desc.) ↑			Goal Acc. ↑
			0.3	0.5	0.7	0.3	0.5	0.7	
EK100†	EK100†	Step	55.32	34.48	15.91	14.13	9.04	4.35	29.71
		Substep	65.98	48.01	25.73	13.54	10.51	6.09	
EgoGS	EK100†	Step	42.49	25.62	12.21	11.24	7.48	3.98	28.26
		Substep	44.91	26.09	11.39	9.81	6.08	2.72	
EgEx†	EK100†	Step	44.39	26.22	10.50	11.43	7.15	3.29	25.36
		Substep	50.75	30.74	13.82	10.68	6.80	3.29	
EgoGS + EgEx†		**Step**	**51.64**	**32.47**	**13.84**	**12.52**	**8.84**	**4.26**	**29.71**
		Substep	**53.67**	**33.98**	**15.98**	**11.97**	**8.11**	**4.24**	

Table 4. Cross-validation results. The EK100-trained model serves as the upper bound, while the EgoGS + EgEx-trained model yields comparable results, demonstrating scalability when merging multiple datasets.

dural videos, where many instances are strictly adjacent.

Comparison with Other Methods As the first work in Hierarchical Streaming Video Understanding, we establish two baseline approaches: (i) We extend recent OAD methods by training class-agnostic variants of [2, 39, 50] using temporal annotations. For detected action instances, we apply the same description generation process as OpenHOUSE. (ii) We compare against SDVC [51], treating the task as an extension of streaming dense video captioning. We introduce two SDVC variants: fine-tuned one (SDVC) following the original implementation and another (SDVC*) that uses only the action proposals from the fine-tuned model while following OpenHOUSE's description generation process with a frozen large-scale VLM.

Table 3 shows OpenHOUSE significantly outperforms both baselines across all metrics, yet comparing the baselines provides additional insights. The higher F1 (loc.) of the OAD extension (TeSTra, MiniROAD, MAT) over SDVC suggests OAD-based approaches are more effective for online localization, supporting our choice of extending OAD-based On-TAL in OpenHOUSE. Additionally, the better performance of SDVC* over SDVC indicates that, for relatively small-scale datasets, leveraging a powerful pretrained VLM may be more effective than full fine-tuning.

Moreover, the goal accuracy gap between baselines and our method demonstrates that accurate lower-hierarchy action predictions are crucial for identifying high-level semantics, as they provide relevant information. Additionally, experiments on the exo-view in EgoExo4D show that OpenHOUSE is not limited to egocentric views, though third-

	Hier.	SDVC [51]		OpenHOUSE
		70	105	
AEDT (s) ↓	Step	36.36	64.84	4.94
	Substep	43.95	52.64	1.82
F1@0.3 (loc.) ↑	Step	9.85	14.12	51.58
	Substep	5.48	8.55	55.17

Table 5. Comparison of prediction delay (AEDT [19]) and F1 (loc.) between SDVC [51] and OpenHOUSE in EgoGS [34]. The numbers 70 and 105 denote the decoding points interval of SDVC.

Num. Frames	Frames Hier.	Previous Output	F1 (loc. + desc.)↑	GPT Score
59385	✗	✗	13.08	2.59
22770	✔	✗	10.96	2.42
22770	✔	✔	13.05	2.76

Table 6. Ablation study on different uses of context memory for generating EgoGS step captions. "Frames Hier." refers to the hierarchical-aware frame sampling strategy, while "Previous Output" refers to using previous predictions as auxiliary information. InternVL2-76B [6] is used across all the experiments.

person view is not the primary focus of this paper.

Cross-Dataset Evaluation Thanks to "class-agnostic" streaming module and VLM's "zero-shot" inference strategy, OpenHOUSE can be trained on a merged dataset and evaluated across different datasets.

For the experiment, we fixed the validation dataset to EK100 while varying the training datasets. Table 4 shows the cross-dataset evaluation results. Comparing the EK100-trained configuration with others, OpenHOUSE demonstrates strong performance even in cross-dataset scenarios. Notably, we observe scalability: training on the merged dataset (EgoGS + EgoEx) yields the highest F1 score. This indicates that OpenHOUSE can be further extended with more training videos, highlighting the strong scalability and potential of the framework.

Prediction Delay To quantitatively assess the promptness of OpenHOUSE, we compare its prediction delay with SDVC using the AEDT metric from [19], which measures the time gap between when a true positive prediction is generated and when it should ideally be produced. Table 5 presents the results, showing that SDVC [51] introduces significant delays compared to our strictly online approach. Moreover, simply shortening the interval between decoding points significantly degrades localization performance, as noted in [51], making it unsuitable for online approaches.

Utilization of Context Memory As discussed in Section 3.3.2, we leverage context memory during VLM inference. The key information in the context memory includes hierarchical frame membership (e.g., used when selecting frames within the substeps of the current step for VLM inference) and previous text-based predictions. Table 6 presents the results of these experiments, showing that incorporating context memory that leverages the hierarchical structure improves performance while requiring only 40% of the frames. This reduction in frame usage significantly lowers the computational complexity of VLM inference.

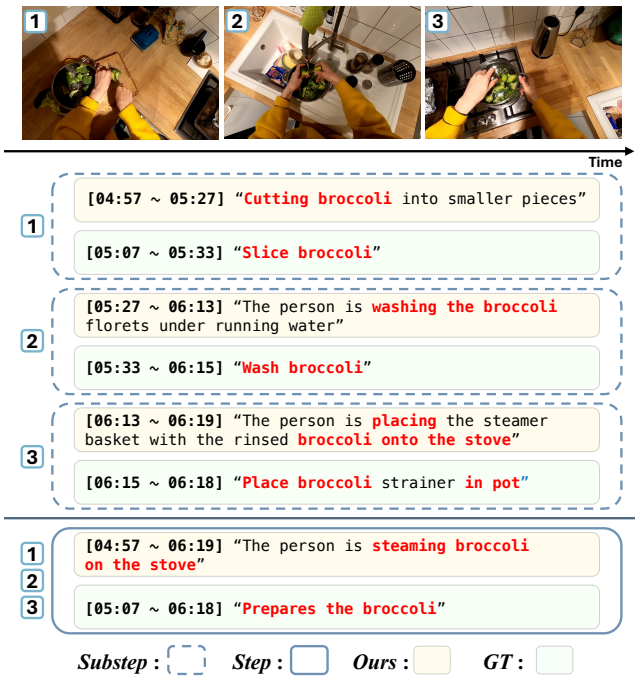


Figure 5. Qualitative results. For the GT substep caption *"Place broccoli strainer in pot"*, OpenHOUSE provides *"The person is placing the steamer basket with the rinsed broccoli onto the stove."*, which aligns better with the actual visual content.

Inference speed We evaluated the inference efficiency on a 46-minute video at 16 fps (totaling 2758 * 16 frames) using four RTX 3090 GPUs with the VLM model InternVL2-40B-AWQ [6]. OpenHOUSE achieves an average processing speed of 24 fps, which is **16 times faster** compared to using the VLM for every frame in the same environment. This shows that our framework significantly improves the efficiency of VLM utilization given the same input.

Qualitative results Figure 5 shows the qualitative results of our OpenHOUSE framework. The video is processed in an online manner, generating both substep and step captions. We observed that the generated captions exhibit high semantic similarity to the ground truth (GT) captions, providing even more detailed descriptions.

Additional experiments Experiments on various VLMs, description generation for incomplete action instances, additional details on the mentioned experiments, and demo videos are provided in the supplementary materials.

5. Conclusion

With advancing VLMs, we see their integration into streaming settings as the next step for online action perception. In this context, OpenHOUSE is a key step, laying the foundation for future research.

Acknowledgements. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2025-00554790), and Artificial Intelligence Graduate School Program, Yonsei University, under Grant RS-2020-11201361.

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. 1, 6

[2] Joungbin An, Hyolim Kang, Su Ho Han, Ming-Hsuan Yang, and Seon Joo Kim. Miniroad: Minimal rnn framework for online action detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 2, 5, 7

[3] Siddhant Bansal, Chetan Arora, and C.V. Jawahar. My view is the best view: Procedure learning from egocentric videos. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[4] Junwen Chen, Gaurav Mittal, Ye Yu, Yu Kong, and Mei Chen. Gatehub: Gated history unit with background suppression for online action detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[5] 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. 2, 3

[6] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 7, 8

[7] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2016. 2

[8] 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. *arXiv preprint arXiv:2006.13256*, 2020. 2, 4, 6

[9] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 6

[10] Hyunjun Eun, Jinyoung Moon, Jongyoul Park, Chanho Jung, and Changick Kim. Learning to discriminate information for online action detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[11] Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taïga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, et al. Stop regressing: Training value functions via classification for scalable deep rl. *arXiv preprint arXiv:2403.03950*, 2024. 5

[12] Ross Girshick. Fast r-cnn. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015. 2

[13] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 4, 6

[14] Ehsan Imani, Kai Luedemann, Sam Scholnick-Hughes, Esraa Elelimy, and Martha White. Investigating the histogram loss in regression. *arXiv preprint arXiv:2402.13425*, 2024. 5

[15] Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. Video recap: Recursive captioning of hour-long videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3, 4

[16] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13700–13710, 2024. 6

[17] Hyolim Kang, Kyungmin Kim, Yumin Ko, and Seon Joo Kim. Cag-qil: Context-aware actionness grouping via q imitation learning for online temporal action localization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 3, 5

[18] Hyolim Kang, Jungsuk Hyun, Joungbin An, Youngjae Yu, and Seon Joo Kim. Actionswitch: Class-agnostic detection of simultaneous actions in streaming videos. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 3, 5, 6

[19] Young Hwi Kim, Hyolim Kang, and Seon Joo Kim. A sliding window scheme for online temporal action localization. In *European Conference on Computer Vision (ECCV)*, 2022. 3, 8

[20] Young Hwi Kim, Seonghyeon Nam, and Seon Joo Kim. 2pesnet: Towards online processing of temporal action localization. *Pattern Recognition*, 2022. 2

[21] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 5

[22] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[23] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024. 6

[24] Ye Liu, Zongyang Ma, Zhongang Qi, Yang Wu, Ying Shan, and Chang W Chen. Et bench: Towards open-ended event-level video-language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 6

[25] 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. 1, 2, 6

[26] Baoqi Pei, Guo Chen, Jilan Xu, Yuping He, Yicheng Liu, Kanghua Pan, Yifei Huang, Yali Wang, Tong Lu, Limin Wang, et al. Egovideo: Exploring egocentric foundation model and downstream adaptation. *arXiv preprint arXiv:2406.18070*, 2024. 6

[27] 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 (ICML)*, 2021. 1, 6

[28] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[29] Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. Timechat: A time-sensitive multimodal large language model for long video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[30] Sakib Reza, Yuexi Zhang, Mohsen Moghaddam, and Octavia Camps. Hat: History-augmented anchor transformer for online temporal action localization. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 3, 6

[31] F. Sener, D. Chatterjee, D. Shelepov, K. He, D. Singhania, R. Wang, and A. Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[32] Yuhan Shen and Ehsan Elhamifar. Progress-aware online action segmentation for egocentric procedural task videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 5

[33] Enxin Song, Wenhao Chai, Guanhong 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. 6

[34] 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*, 2024. 2, 3, 4, 6, 8

[35] Youngkil Song, Dongkeun Kim, Minsu Cho, and Suha Kwak. Online temporal action localization with memory-augmented transformer. In *European Conference on Computer Vision (ECCV)*, 2024. 2, 3, 6

[36] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[37] Tuan N Tang, Jungin Park, Kwonyoung Kim, and Kwanghoon Sohn. Simon: A simple framework for online temporal action localization. *arXiv preprint arXiv:2211.04905*, 2022. 2, 3

[38] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6

[39] Jiahao Wang, Guo Chen, Yifei Huang, Limin Wang, and Tong Lu. Memory-and-anticipation transformer for online action understanding. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 7

[40] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Zhengrong Zuo, Changxin Gao, and Nong Sang. Oadtr: Online action detection with transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[41] 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. 3

[42] 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. 6

[43] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S Davis, and David J Crandall. Temporal recurrent networks for online action detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[44] 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 (NeurIPS)*, 2021. 2

[45] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10714–10726, 2023. 6, 7

[46] Le Yang, Junwei Han, and Dingwen Zhang. Colar: Effective and efficient online action detection by consulting exemplars. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[47] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 7

[48] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision (ECCV)*, 2022. 6

[49] Qingsong Zhao, Yi Wang, Jilan Xu, Yinan He, Zifan Song, Limin Wang, Yu Qiao, and Cairong Zhao. Does video-text pretraining help open-vocabulary online action detection? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 2

[50] Yue Zhao and Philipp Krähenbühl. Real-time online video detection with temporal smoothing transformers. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 7

[51] Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. Streaming dense video captioning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3, 6, 7, 8

[52] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 6