Progress-Aware Video Frame Captioning

Zihui Xue, Joungbin An, Xitong Yang*, Kristen Grauman
UT Austin

*

Abstract

While image captioning provides isolated descriptions for individual images, and video captioning offers one single narrative for an entire video clip, our work explores an important middle ground: progress-aware video captioning at the frame level. This novel task aims to generate temporally fine-grained captions that not only accurately describe each frame but also capture the subtle progression of actions throughout a video sequence. Despite the strong capabilities of existing leading vision language models, they often struggle to discern the nuances of frame-wise differences. To address this, we propose ProgressCaptioner, a captioning model designed to capture the fine-grained temporal dynamics within an action sequence. Alongside, we develop the FrameCap dataset to support training and the FrameCapEval benchmark to assess caption quality. The results demonstrate that ProgressCaptioner significantly surpasses leading captioning models, producing precise captions that accurately capture action progression and set a new standard for temporal precision in video captioning. Finally, we showcase practical applications of our approach, specifically in aiding keyframe selection and advancing video understanding, highlighting its broad utility. [1]

1. Introduction

Visual captioning [38]—the task of generating textual descriptions of visual content—is a fundamental problem in computer vision with extensive practical applications. Existing captioning paradigms are broadly divided into two categories: image captioning and video captioning, with a clear distinction between them. Image captioning [23] generates a single, isolated description for each image, with no contextual linkage among different images. In contrast, video captioning [1] assigns a single caption for the entire video clip, aggregating information across frames without addressing the specifics of individual frames.

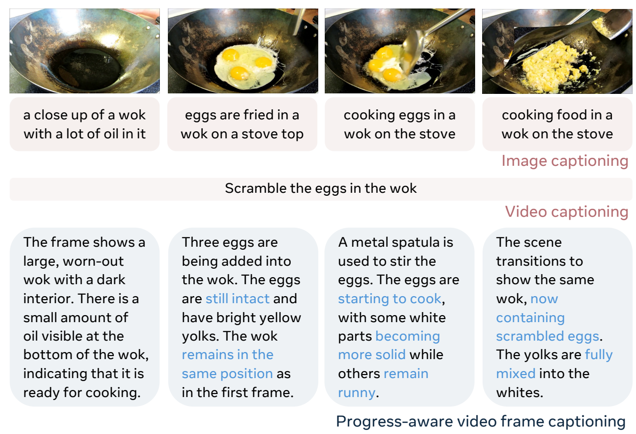


Figure 1. We propose progress-aware video frame captioning (bottom), which aims to generate a sequence of captions that capture the temporal dynamics within a video. Unlike traditional image and video captioning (top) that focus on broad event-level descriptions, our task delves into the detailed, progressive dynamics of an action, necessitating precise, temporally fine-grained capabilities. Blue text highlights how the progress-aware captions build successively on the earlier content to highlight what is changing.

Figure 1 illustrates this dichotomy. Employing an image captioning model like BLIP [37] to describe each frame of the video results in captions that are *local, not temporally context-aware*, and may exhibit little variation across the sequence. Conversely, video captioning provides a *global, not temporally fine-grained* overview of the event, as exemplified by the YouCook2 [84] ground truth label "scramble the eggs in the wok". In both scenarios, the nuances of how the action unfolds over time are missed. This raises the question: Can we develop temporally fine-grained captions that capture the subtle, progressive nature of action sequences? Figure 1 (bottom) illustrates what we seek.

Having such progress-aware captions could benefit a great variety of downstream tasks, bringing improved video understanding [73, 78], more precise video retrieval [66–68], and enriched video generation [50, 58]. Moreover, such a capability could open up new AR/VR and robotics applications. For instance, in AI coaching, a captioning system could meticulously analyze an expert's tennis forehand, simplifying the learning process for users. Similarly,

*Work conducted as an independent researcher.

[1] Project webpage: https://vision.cs.utexas.edu/projects/ProgressCaptioner.

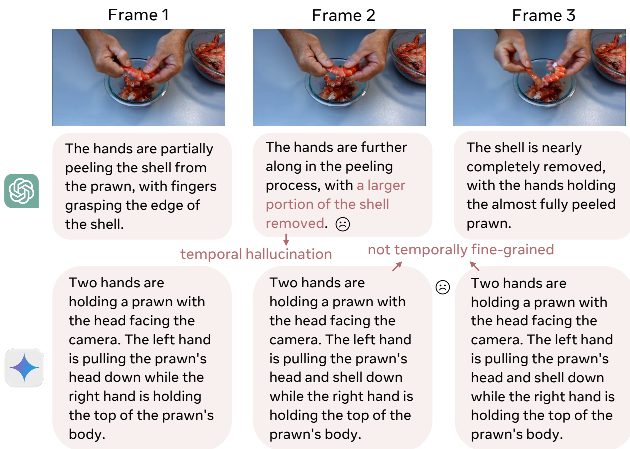


Figure 2. Issues of existing VLMs in video frame captioning: (1) Lack of temporal granularity. See captions for frames 2 and 3, produced by Gemini-1.5-Pro (row 2), which fail to distinguish subtle differences between the frames. (2) Temporal hallucination. See frame 2's caption produced by GPT-4o (row 1), which inaccurately suggests progression that is not visible.

for how-to video creation, it could elicit and describe the key object state changes at each stage (e.g., "how to make whipped cream")—useful for both content creators as well as visually-impaired users learning a new skill.

Towards this end, we introduce a novel captioning task—*progress-aware video captioning at the frame level*. This task involves generating captions that not only coherently depict action progression but also tailor each description specifically to its corresponding frame.[2] Our task is uniquely characterized by its demand for fine-grained temporal sensitivity. By "fine-grained", we refer to generating detailed, frame-level descriptions that elucidate the stages or procedural steps of the action, effectively conveying how the action is performed throughout the video sequence.

As discussed, traditional video captioning [9, 63, 70, 85] settles for broad event-level descriptions, where a description like scrambling eggs for the video in Figure 1 would be considered entirely accurate. In contrast, we seek progress-aware captions that detail each stage of the action, such as "eggs still intact", "starting to cook" and "fully mixed". While recent works [6, 8, 11, 12, 61, 82] enhance the overall descriptiveness of video captions, they continue to produce a single video-level description without distinguishing the nuances between individual frames. Our task delves deeper, exploring how each frame contributes to the narrative of the action's progression, thereby setting a new standard for fine-grained temporal precision in video captioning.

Despite great advancements of vision language models (VLMs) [2, 33, 37, 40–42, 62, 75, 80, 82] that have markedly improved visual captioning, we observe that these models still struggle with this nuanced task. Two main issues persist: first, the lack of temporally fine-grained captions; when shown adjacent frames that depict subtle variations in action progression (such as frames 2 and 3 in Figure 2), the generated captions can be overly coarse, failing to differentiate between the frames (see row 2, Gemini-1.5-Pro's captions). Second, we identify and term a notable issue of "temporal hallucination", where the captions suggest temporal progression in disagreement with what the visual frames exhibit. See frame 2 of Figure 2, where GPT-4o's generated captions (row 1) incorrectly advance the action sequence. The prevalence of such errors can be attributed to models' reliance on the common statistics of activity sequences, which mistakenly override matching specific statements to specific frames. Meanwhile, image captioners—even if trained with fine-grained annotations—treat frames in isolation and hence lack the temporal context to say what is *progressing* versus what is *present*.

We propose ProgressCaptioner, a captioning model designed to generate progress-aware frame-level captions. This model is developed through an innovative strategy that interleaves pseudo labeling with two learning stages. In the first stage, we employ multiple VLMs to generate consensus pseudo captions for *pairs* of frames, and we devise two novel evaluation tasks to automatically assess their quality. In the second stage, armed with that initial model, we apply a two-frame sliding window to generate more precise captions across the full frame sequence, and we generalize our custom evaluation tasks to identify high-quality caption sequences. The final model, ProgressCaptioner, is then trained on this comprehensive dataset, which we term FrameCap, to excel in producing temporally fine-grained captions that accurately characterize action progress.

To assess the quality of frame-wise captions and benchmark ProgressCaptioner against leading VLMs, we introduce the FrameCapEval benchmark comprised of videos from four public video action datasets. ProgressCaptioner consistently outperforms leading open-source VLMs with a 1.8× to 2.7× improvement in caption quality and also achieves the highest selection rate in user studies, even surpassing the much larger proprietary models GPT-4o [2] and Gemini-1.5-Pro [53]. Finally, we highlight potential applications enabled by our advanced captions: keyframe selection and enhanced video understanding. We hope that our task, model, and benchmark can inspire future development in temporally fine-grained video captioning.

2. Related Work

Image Captioning Image captioning has been extensively studied in recent years [3, 13, 60, 76]. A related line of work is image difference captioning, where the task is to

[2]Without loss of generality, we obtain the input frame sequence by uniformly sampling from the action clips at a fixed rate (1FPS). These frames may or may not demonstrate visual action progression from one to the next, demanding that the model discerns the difference when generating progress-aware captions.

describe differences between two images [27, 48, 56] or sets of images [15]. Building on the success of generative models, recent benchmarks [4, 28, 77] challenge models to distinguish between two visually similar images, advancing fine-grained image understanding. However, all the above models are restricted to static (typically synthesized) image pairs and address coarse-grained differences like object presence or absence. Temporal intricacies—accurately describing how an action progresses—remain unexplored.

Video Captioning Video captioning [1] aims to produce a single description that encapsulates a video clip. While traditional benchmarks [9, 63, 70, 85] offer a brief one-sentence caption for each video, recent efforts expand this scope, extending captioning to hours-long videos [26], enriching the granularity of details [8, 11, 61], enhancing caption uniqueness [49], integrating a casual temporal narrative [46], or introducing LLM summarization [34].

Adjacent to traditional video captioning, are the tasks of visual storytelling [25, 36] (creating a coherent story for a sequence of snapshots), dense video captioning [30, 74, 86] (temporal localization and captioning of all events in an untrimmed video), and audio description [20–22] (detailed narrations of visual events in videos (e.g., movies) for visually impaired audiences). However, all these works still address "what is happening" at a coarse-grained event level, e.g., noting that someone is making a souffle within a specific time range. The ability to break down frame-level details—such as whisking egg whites, folding ingredients, and observing the souffle rising—is still lacking.

Vision Language Models Recent advancements in VLMs [2, 33, 37, 40–42, 62, 75, 80, 82] have greatly enhanced the capabilities of both image and video captioning. Despite their strong performance, VLMs often exhibit "hallucination" [19, 65], and preference learning [52, 83] has proven effective in mitigating this issue.

Compared to image-LMs, video-LMs crucially require the integration of temporal dynamics understanding, spurring a series of work on evaluating temporal perception [18, 39, 43, 69]. While these assessments ensure that a model can generate an accurate overall video summary or answer general questions, they entail neither temporal localization nor discernment of fine-grained differences between frames. The OSCaR benchmark [47] focuses on object state change (OSC) captioning, yet it is limited to just three frames and specifically OSC videos, with models and captions not publicly released yet preventing direct comparison. Additionally, their approach relies on human annotation and a single advanced GPT model. In contrast, our approach features a scalable data collection pipeline that reduces reliance on these labor-intensive resources, employs novel automatic evaluation tasks, and broadens the scope beyond OSC videos. Finally, unlike methods for long-form video and event localization with VLMs [10, 44, 54], our focus is distinctly more temporally fine-grained, concentrating on how individual frames evolve within a single event.

3. Approach

We delve into the specific challenges of our progress-aware video frame captioning problem in Sec. 3.1 and outline ProgressCaptioner's development in Sec. 3.2.

3.1. Progress-aware Video Frame Captioning

Problem Formulation Our objective is to develop a captioning model that, given a video, produces accurate and temporally fine-grained captions. Formally, for a sequence of T frames, denoted as $\mathcal{V} = \{v_i\}_{i=1}^T$, the captioning model generates a corresponding sequence of captions $\mathcal{C} = \{c_i\}_{c=1}^T$, where each c_i describes the i-th frame v_i. This captioning process is underpinned by three key requirements: (1) *Accuracy*, where each caption c_i must faithfully represent what is visually occurring in frame v_i, without hallucinating from the context of other frames; (2) *Temporal Specificity*, where each caption c_i specifically attends to v_i, without being overly generic to be applicable to multiple frames in the sequence; (3) *Progressive Coherence*: The sequence of captions $\{c_i\}_{i=1}^T$ should build upon each other to reflect the essential changes in the action over time.

FrameCap Dataset To train our captioning model, we require a dataset that pairs frame sequences ($\mathcal{V}$) with corresponding captions ($\mathcal{C}$). Existing datasets [9, 63, 70, 85] provide only a single, generic caption for an entire video clip, lacking the frame-wise caption format we need. To address this gap and train our model, we develop the FrameCap dataset. Given the prohibitive expense of collecting human-labeled caption sequences as our ground truth ($\mathcal{C}$), especially at scale, we leverage leading VLMs as powerful tools to create a pseudo caption sequence $\hat{\mathcal{C}}$ from $\mathcal{V}$. For video sources, we refer to two large-scale datasets that focus on fine-grained human activities: HowToChange [72] (featuring object state change videos from YouTube) and COIN [59] (featuring daily activities from YouTube).

Caption Sequence Construction Prompting VLMs for our desired caption sequence is nontrivial. We identify two key problems: (1) *Input considerations*: how many context frames from $\{v_i\}_i^T$ should be provided? (2) *Output assessment*: what issues arise in VLM-generated captions, and how can we filter to retain only high-quality ones? To explore these questions, we conduct preliminary experiments by prompting leading VLMs to perform the frame-wise captioning task. We share our findings below.

Observation I Intuitively, inputting all T frames would seem best. However, current VLM capabilities do not support this extensive context. Specifically, providing too many frames at once often leads to descriptions that lack detail and exhibit temporal inaccuracies, with VLMs also risk-

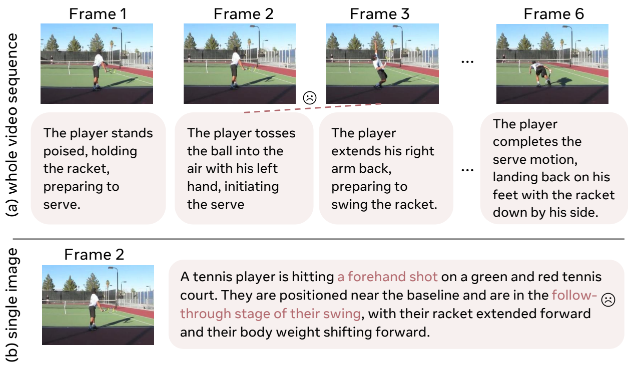


Figure 3. Captioning outcomes using Gemini-1.5-Pro [53].

ing memory overload, as similarly observed in [11]. Conversely, providing a single frame at a time reduces the task to image captioning, which is not optimal either, resulting in captions that lack temporal context and coherence.

Figure 3 shows a representative trial with Gemini-1.5-Pro [53]. Inputting the full sequence (case (a)) yields brief per-frame descriptions with temporal misalignment (i.e., the second caption erroneously describes what is visually occurring in the third frame). On the other hand, captioning frames in isolation (case (b)) removes essential temporal context, where the model mistakes the initial stage of a tennis serve for the follow-through of a forehand swing. These findings underscore the importance of finding a balanced approach and motivate us to adopt a *frame pair* as the stepping stone of our captioning model development.

Observation II Next, building upon the use of a frame pair (v_1, v_2), is the caption pair $(\hat{c}_1, \hat{c}_2)$ produced by existing VLMs of sufficient quality to be directly adopted? Our preliminary experiments reveal two main issues: (a) lack of temporal granularity, and (b) temporal hallucination, as showcased in Figure 2. To dissect these issues, we analyze the captions in relation to the visual progression between frames v_1 and v_2. Specifically, if there is a visible progression from v_1 to v_2 (e.g., the slight peeling of a shrimp's shell from frame 2 to frame 3 in Figure 2), the captions should adequately reflect this change. Overly similar captions in such a scenario signify a failure in temporal granularity. Conversely, when there is no change between frames (e.g. frames 1 and 2 in Figure 2), the captions should remain consistent. We deem it a temporal hallucination when captions erroneously indicate progression in disagreement with the visuals. This issue often arises in VLMs equipped with strong language decoders, which can introduce a language bias toward generating unwarranted narrative changes.

3.2. ProgressCaptioner

The observations above drive the design of our model, ProgressCaptioner, which unfolds in two stages. Based on our findings that current VLMs have trouble maintaining caption quality when handling extensive T-frame inputs, our approach begins with frame pair captioning. In the first stage, we develop a ProgressCaptioner to excel at describing the nuances between adjacent frames. The second stage then leverages the first-stage model to pseudo label the full T-frame sequence with a two-frame sliding window. This staged approach refines caption quality along with model development, enhancing the captioning process iteratively with more precise pseudo labels.

Frame Pair Data Preparation Starting with a frame pair $\mathbf{v} = (v_1, v_2)$, we employ K captioning models to generate an initial set of caption pairs $\{(\hat{c}_1, \hat{c}_2)\}_1^K$. Acknowledging the potential inaccuracies in these captions, as per observation II, we design two automatic evaluation tasks to assess caption quality. The first task, *progression detection*, examines progress awareness: it checks whether the captions appropriately reflect visual changes between v_1 and v_2. Specifically, an LLM assesses each caption pair $(\hat{c}_1, \hat{c}_2)_k$ to determine if they suggest a visible physical change. We utilize majority voting across multiple LLMs' assessments for all K caption pairs to establish a consensus visual-change label. Caption pairs that align with this consensus are marked as passing; others are marked as failing.

For pairs passing progression detection, we proceed to our second evaluation task—*caption matching*—to assess how precisely $\hat{c}_1$ and $\hat{c}_2$ describe v_1 and v_2, respectively. The task is designed as a multi-choice question format, where a VLM is given $\hat{c}_1$, $\hat{c}_2$, and an "unsure" option, and tasked with matching the correct caption to each frame. A caption pair is considered good if the evaluation VLM correctly identifies $\hat{c}_1$ for v_1 and $\hat{c}_2$ for v_2. Because the captions will all be topically related, this is essentially a matching task with "hard negatives" that lets us automatically gauge the precision of the proposed captions for the target images.

This automatic pipeline distinguishes between high-quality caption pairs, denoted by $\hat{\mathbf{c}}^+ = (\hat{c}_1^+, \hat{c}_2^+)$, and those that exhibit inaccuracies or hallucinations, denoted by $\hat{\mathbf{c}}^- = (\hat{c}_1^-, \hat{c}_2^-)$, forming training data for Stage I.[3]

Stage I Training Following the success of versatile VLMs in captioning tasks [34, 42, 61, 71], we initialize ProgressCaptioner with the LLAVA-OV-7B [33] checkpoint to inherit its pretrained capabilities. Stage-I training utilizes frame and caption pair data collected on HowToChange and COIN YouTube videos $<\mathbf{v}, \hat{\mathbf{c}}^+, \hat{\mathbf{c}}^->$ through two principal methods: supervised fine-tuning (SFT) and direct preference optimization (DPO). The SFT process is straightforward given our dataset; we perform instructional tuning to tailor the general capabilities of the original VLM to our specific frame-wise captioning requirements using $<\mathbf{v}, \hat{\mathbf{c}}^+>$. The subsequent DPO step targets the prevalent issue of hallucination in VLMs and is innovatively

[3]We encourage readers to view data examples provided in Supp. for a better understanding of our data refinement process, as well as details on pseudo labeling, prompts used, and the K VLMs we employ.

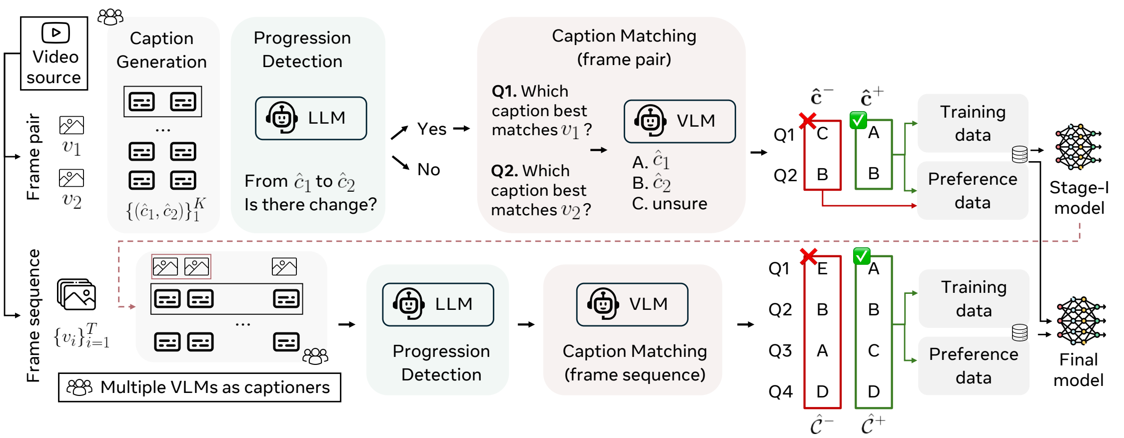


Figure 4. Framework of ProgressCaptioner, designed in two stages. In Stage-I, we prepare frame pairs and generate corresponding caption pairs using multiple VLMs. Each pair undergoes our designed progression detection and caption matching evaluations, to decide if they are selected for model supervised fine-tuning or rejected, with the latter contributing to preference data to aid in model preference learning. The Stage-I model training then proceeds using this collected data. In Stage-II, the trained stage-I model labels frame sequences with a two-frame sliding window, in conjunction with other VLMs. These sequences are again assessed through progression detection and caption matching to classify them as selected or rejected. All collected data from both stages contribute to the final training of ProgressCaptioner.

driven by our proposed automatic evaluation critics. Preference optimization [52] in LLM training typically requires human-provided preference data to steer LLM responses towards more desirable outputs. Here, we employ progression detection and caption matching to automatically construct preference data $\hat{\mathbf{c}}^+$ and $\hat{\mathbf{c}}^-$, eliminating the reliance on manual labeling. This preference data $<\mathbf{v}, \hat{\mathbf{c}}^+, \hat{\mathbf{c}}^->$ is adopted in DPO training to further enhance model performance with feedback from LLM and VLM evaluations.

Frame Sequence Data Preparation The second stage expands our pseudo labeling scheme from 2 to T frames, where our Stage-I ProgressCaptioner first generates captions using a two-frame sliding window. To increase data diversity and volume, we also incorporate captions produced by other VLMs, with both two-frame and full T-frame contexts, since captions of low-quality are also useful (after undergoing our evaluation tasks, those that are rejected enrich the preference data). Once the initial set of caption sequences is generated, we conduct progression detection to identify M visually distinct frames from the original T-frame sequence, denoted as $\mathcal{V}_M = \{v_i\}_{i=1}^M$, using majority voting; M varies based on the distinctiveness of each frame sequence's content. The caption matching task is then employed to encompass M frames, with a selection pool of all M captions, $\hat{\mathcal{C}}_M = \{\hat{c}_i\}_{i=1}^M$, plus an "unsure" option. A high-quality caption sequence $\hat{\mathcal{C}}^+$ is identified when the evaluation VLM correctly selects $\hat{c}_i$ for v_i across all frames. Conversely, a caption sequence is deemed problematic, $\hat{\mathcal{C}}^-$, if the VLM incorrectly answers more than half of the caption selections. This process forms our Stage-II data.

Stage II Training Following the same pipeline as stage I, ProgressCaptioner is first trained through SFT using data prepared during both stages, which includes frame-caption pairs $<\mathbf{v}, \hat{\mathbf{c}}^+>$ and frame-caption sequences $<\mathcal{V}, \hat{\mathcal{C}}^+>$. Subsequently, we conduct DPO with preference data collected from both stages $<\mathbf{v}, \hat{\mathbf{c}}^+, \hat{\mathbf{c}}^->$ and $<\mathcal{V}, \hat{\mathcal{C}}^+, \hat{\mathcal{C}}^->$ to further refine model performance and mitigate hallucination. This sequential approach results in our final captioning model. The framework is illustrated in Figure 4.

4. Experiments

We tackle two questions below: (1) How to evaluate frame-wise caption quality of existing models? And how does ProgressCaptioner perform? (Sec. 4.1); (2) What applications are enabled by precise progress-aware captions? (Sec. 4.2)

4.1. Benchmarking Video Frame Captioning

To systematically evaluate the quality of generated video frame-wise captions, we propose the FrameCapEval benchmark. Our benchmark sets a new standard for temporal precision in video captioning, assessing captions based on the criteria established in Sec. 3.1.

Evaluation Metrics We employ the automatic evaluation tasks of progression detection and caption matching (Sec. 3.2), reporting accuracy with Llama-3.1-70B-Instruct [14] as the evaluation LLM and Gemini-1.5-Pro [53] as the evaluation VLM (see Supp. for details). Additionally, we enhance our evaluation with a **user study** of 15 participants, reporting the average selection rate.

Benchmark Data Curation We establish the FrameCapEval benchmark, featuring videos from **four action understanding datasets**: HowToChange [72] and COIN [59] (on which ProgressCaptioner was trained), along with Penn Ac-

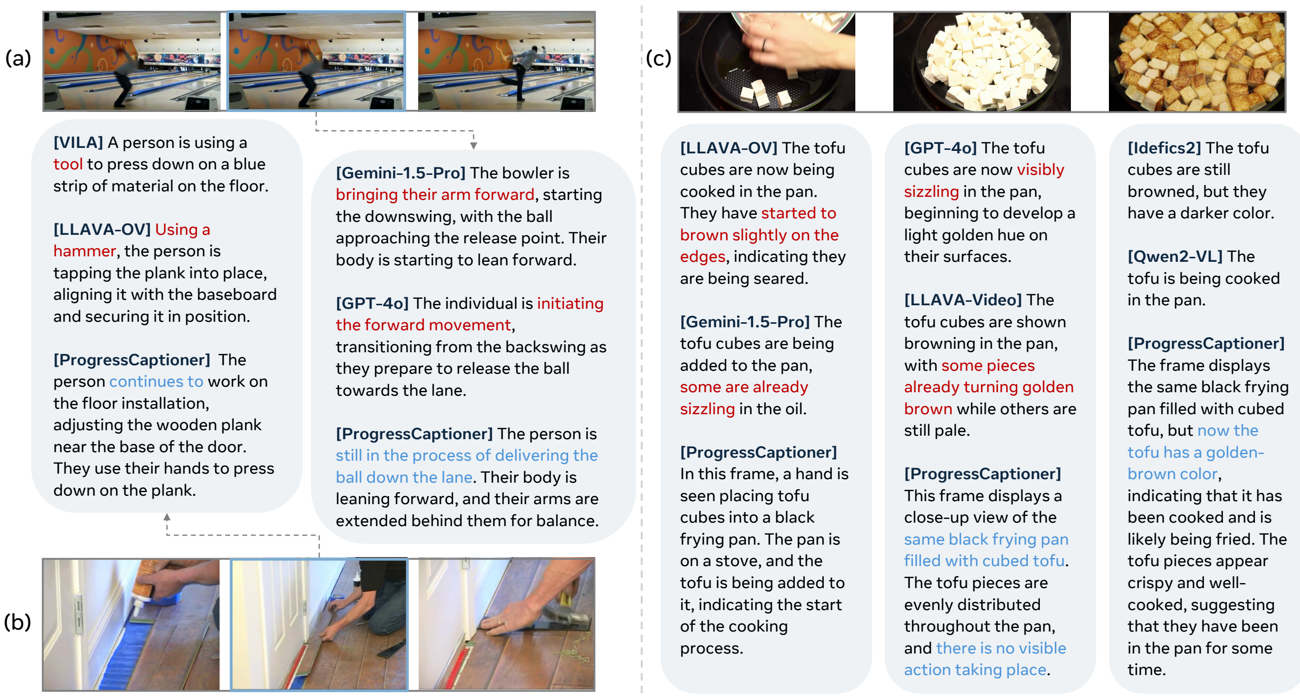


Figure 5. Qualitative comparisons of ProgressCaptioner with SOTA VLMs on three action sequences. For sequences (a) and (b), only the middle frame predictions are displayed. See Supp. for all models' predictions on full sequences and more examples. Inaccuracies in descriptions are highlighted in red. Even top VLMs often produce descriptions that misalign with the corresponding frames, while ProgressCaptioner delivers hallucination-free and temporally fine-grained captions, including phrases explicitly calling out progress (blue).

Model	Size	HTC		COIN		Penn&K	
		Cap	Prog	Cap	Prog	Cap	Prog
Proprietary models							
Gemini-1.5-Pro [53] (img)	-	28.4	59.7	*24.3*	58.6	15.3	51.2
Gemini-1.5-Pro [53]	-	*31.4*	63.8	**25.0**	*63.8*	*17.6*	*60.3*
GPT-4o [2]	-	**32.4**	*64.2*	21.3	58.4	**18.2**	**63.2**
Open-source models							
Idefics2 [31]	8B	2.0	54.4	2.9	52.2	12.5	50.9
VILA [40]	8B	6.9	53.6	5.1	48.2	15.9	51.4
Qwen2-VL [62]	7B	13.7	**69.6**	11.0	**70.8**	8.5	58.8
LLAVA-Video [82]	7B	3.9	59.3	8.8	53.0	9.7	51.8
LLAVA-OV [33] (img)	7B	5.9	56.3	17.6	55.4	11.9	55.5
LLAVA-OV [33]	7B	7.8	59.0	5.9	57.3	5.1	50.8
ProgressCaptioner (ours)	7B	**37.3**	**73.6**	**32.3**	**66.1**	**31.3**	**63.7**

Table 1. Results on the FrameCapEval Benchmark, composed of video from four public datasets. Cap and Prog denote caption matching and progression detection accuracy, respectively. ProgressCaptioner greatly outperforms SOTA open-source VLMs and even the leading proprietary models, despite being a 7B model. The **best** results are bolded and underlined, the **second best** are bolded, and the *third best* are italicized. Moreover, the results confirm our model's generalizability from in-domain datasets (HTC for HowToChange and COIN) to external datasets not seen during training (Penn&K, representing Penn Action and Kinetics).

tion [81] and Kinetics [7], which are unseen in training and serve to assess generalization capabilities. We are mindful

Figure 6. User study results comparing ProgressCaptioner with top competitors show it as the most preferred model (see text).

of the single frame bias [32] and manually verify all videos to exclude sequences lacking fine-grained action progression. This process yields a final set of 684 videos.

Baselines We evaluate an array of state-of-the-art VLMs, including two proprietary models, GPT-4o [2] and Gemini-1.5-Pro [53], and five open-source models—Idefics2 [31], VILA [40], Qwen2-VL [62], LLAVA-Video [82], and LLAVA-OV [33], all of which process the entire T-frame sequences as input. To ensure a comprehensive evaluation, we also incorporate image captioning baselines using Gemini-1.5-Pro and LLAVA-OV. We select open-source VLM variants with fewer than 10B parameters for computational efficiency and a fair comparison with our ProgressCaptioner, which is an 7B model. The closed proprietary

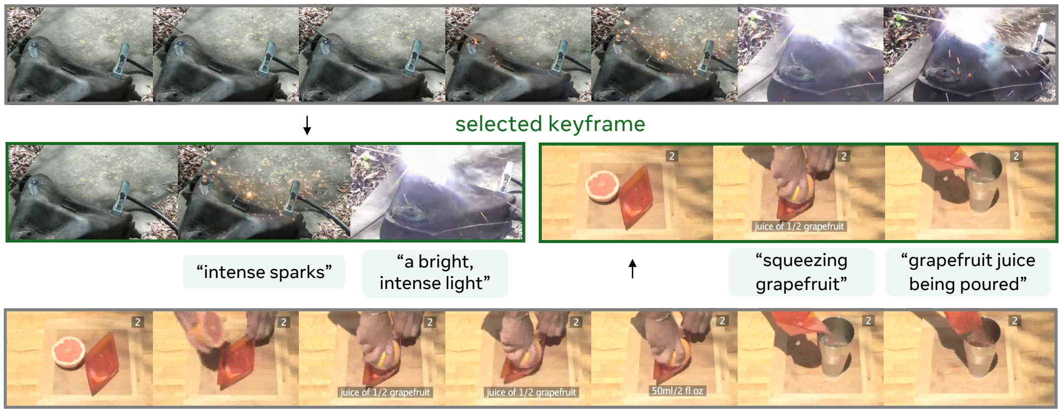


Figure 7. ProgressCaptioner facilitates keyframe selection and enriches the selected keyframes with progress-aware descriptions.

models are much larger and trained with much more extensive data; we include them as a useful reference point, but stress that they do not constitute an apples-to-apples comparison, to the disadvantage of our ProgressCaptioner.

Implementation ProgressCaptioner is constructed with SigLIP [79] as the vision encoder and Qwen2 [62] as the language model, linked through a projector, and initialized from the LLAVA-OV-7B checkpoint [33]. For final-stage model training, we finetune ProgressCaptioner on 274.5K data samples over 1 epoch during the SFT stage and on 47.3K preference data for 1 epoch during the DPO stage.

Results As shown in Table 1, on the FrameCapEval benchmark, ProgressCaptioner greatly outperforms existing open-source VLMs of similar capacity and even the (much larger) latest Gemini-1.5-Pro and GPT-4o. We observe that strong language-backed VLMs like GPT-4o show high caption matching accuracy, whereas Qwen2-VL excels in progression detection, reducing hallucination. However, it tends to produce less detailed captions, leading to lower caption matching accuracy. In contrast, ProgressCaptioner effectively balances precision and detail in frame-wise captioning, consistently leading the benchmark across both in-domain and out-of-domain datasets.

Figure 5 provides qualitative comparisons on three action sequences. Consider the (a) bowling sequence for instance: baseline models erroneously suggest progression in frame 2, like "arm forward", exemplifying the common issue of temporal hallucination in current VLMs. This issue recurs in the other two sequences. Conversely, ProgressCaptioner delivers high-quality captions that precisely characterize action progress in each frame. See Supp. for more qualitative examples and an ablation of ProgressCaptioner.

User Study Figure 6 presents the user study results, where ProgressCaptioner is compared against four of the strongest competitors: LLAVA-OV, Qwen2-VL, Gemini-1.5-Pro and GPT-4o. Each participant is presented with five captions produced by these models and is tasked with selecting the top-2, with an additional "none" option if the captions are deemed inadequate. ProgressCaptioner emerges as the most preferred model, with an average best caption selection rate of 31.6%—2× to 3.6× better than the comparably sized best models from the literature [33, 62], and even surpassing top-tier proprietary models that enjoy significant scaling advantages. While our model outperforms all open-source *and* proprietary models for top-1 preference, the more forgiving top-2 metric brings the proprietary closed models back in the game, though our model remains competitive even there (50.3% for GPT-4o vs. 47.3% for ours). These findings underscore our model's strong ability to produce accurate, temporally fine-grained captions.

4.2. Applications of Video Frame Captioning

ProgressCaptioner offers progress-aware frame-wise captions, which hold great potential for many real-world applications. We explore several practical use-cases below.

Keyframe Selection Our first use-case leverages ProgressCaptioner's temporally precise captions as an intermediate representation to identify keyframes within a densely sampled sequence, aided by an LLM (see Supp for details). Figure 7 provides two examples, showcasing how ProgressCaptioner's produced captions allow selecting distinct frames that effectively capture different stages of the welding and squeezing grapefruit action. While recent video summarization work [24] explores using VLMs and LLMs for keyframe selection, it aims to identify coarse-grained events within long videos, which is not adequate for our problem scenario. See Supp. for a side-to-side comparison and more qualitatives that underscore this distinction.

Keyframes for Action Recognition Not only is keyframe selection useful for human viewers to quickly preview a longer video, but it can also extract the most informative portions of a video to benefit activity recognition [29]. To illustrate, we next apply our keyframe selection mecha-

Figure 8. On Kinetics test videos, ProgressCaptioner selects four frames that are more informative of the action than uniform sampling, resulting in improved action recognition accuracy.

nism to Kinetics [7] Temporal [57] subset that necessitates multi-frame reasoning. Given that the original video clips are short (sampled at 1FPS, resulting in sequences of 10 frames), we employ two models that take four frames as input: Slow backbone from SlowFast [17] and X3D-XS [16]. We prompt GPT-4o to select four representative frames from frame-wise captions produced by ProgressCaptioner. We take the two model checkpoints that have been trained on the Kinetics training set and replace uniformly sampled frames with our selected keyframes during inference. Figure 8 presents a qualitative comparison, highlighting performance gains such as a +1.7% increase in top-1 accuracy for both SlowFast and X3D models. Even among just 10 candidate frames, our method's fine-grained ability to identify the 4 most informative ones translates into better recognition.

Advancing Video Understanding The precise, frame-wise captions generated by ProgressCaptioner enhance our understanding of videos. To demonstrate this, we consider two video tasks that demand temporally fine-grained understanding: (1) frame-wise classification on HowToChange [72] and Penn Action [81], and (2) video question answering (QA) on NExT-QA [69] (ATP-Hard [5]). These tasks are chosen because they challenge the model to comprehend not just the overarching content of a video, but also the more fine-grained event progression within a video. The HowToChange and Penn Action test sets provide frame-wise labels detailing object state changes or action phases, requiring frame-level understanding. Similarly, NexTQA (ATP-Hard) poses temporally challenging questions that demand multi-frame reasoning, such as determining event order, emphasizing the need for precise temporal comprehension. For baseline comparisons, we evaluate against the LLAVA-OV-7B [33] model, from which ProgressCaptioner is initialized, to highlight the enhancements that our specialized training on FrameCap brings to video understanding tasks. For the first task, as we pioneer a zero-shot, language-guided approach to this traditionally vision-centric problem (details below), no other zero-shot baselines exist. For the second task, we compare ProgressCaptioner against two existing zero-shot approaches [45, 64].

Zero-shot Frame Classification We repurpose zero-shot frame-wise classification task into a multi-choice QA format, using frame-wise captions to guide an LLM in identifying the correct label per frame, evaluating caption accuracy and granularity (Figure 9 left). Results (Figure 9 right) show that ProgressCaptioner consistently outperforms LLAVA-OV across both datasets. Notably, our training involves no signals related to these frame-wise labels, underscoring its generalizability and effectiveness in enhancing video frame-level understanding.

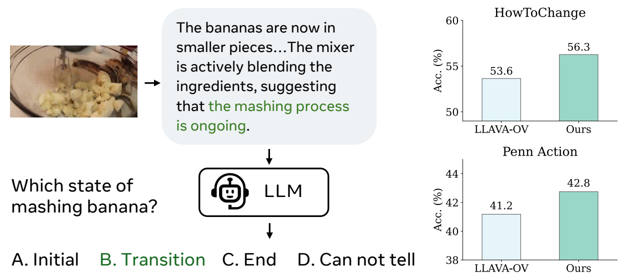


Figure 9. ProgressCaptioner delivers precise and detailed per-frame descriptions, leading to enhanced zero-shot frame-wise classification performance when compared with LLAVA-OV.

Video QA Finally, we report results using frame-wise descriptions for video QA, where an LLM (we use GPT-4o) is employed to answer questions on NExT-QA (ATP-Hard) set. As shown in Table 2, ProgressCaptioner achieves the best results on this benchmark, outperforming the previous leader VideoAgent [64] by +3.4%. Compared with a similar setup using LLAVA-OV, ProgressCaptioner achieves a +4.7% gain in the temporal subset, highlighting its superior ability to produce fine-grained, temporally precise descriptions and bring enhanced video understanding.

Model	Acc@C	Acc@T	Acc@All
VFC [45]	32.2	30.0	31.4
VideoAgent [64]	57.8	58.8	58.4
LLAVA-OV [33] + GPT-4o	62.6	53.4	58.8
ProgressCaptioner + GPT-4o (ours)	64.4	58.1	**61.8**

Table 2. Video QA results on NExT-QA (ATP-Hard). C and T denote causal and temporal subsets, respectively.

5. Conclusion

We introduce progress-aware video frame captioning, which necessitates a significant enhancement in current captioning models' capability to describe temporal action progression precisely. Towards this end, we develop ProgressCaptioner and show its effectiveness in enhancing the temporal precision and alignment of captions with corresponding frames. Furthermore, we demonstrate its practical applications: keyframe selection and enhanced video understanding. By setting a new standard for temporal precision in video captioning, we hope our work inspires further development in this evolving domain.

References

[1] Moloud Abdar, Meenakshi Kollati, Swaraja Kuraparthi, Farhad Pourpanah, Daniel McDuff, Mohammad Ghavamzadeh, Shuicheng Yan, Abduallah Mohamed, Abbas Khosravi, Erik Cambria, et al. A review of deep learning for video captioning. *arXiv preprint arXiv:2304.11431*, 2023. 1, 3

[2] 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, 3, 6

[3] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8948–8957, 2019. 2

[4] Rabiul Awal, Saba Ahmadi, Le Zhang, and Aishwarya Agrawal. Vismin: Visual minimal-change understanding. *arXiv preprint arXiv:2407.16772*, 2024. 3

[5] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the "video" in video-language understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2917–2927, 2022. 8

[6] Mu Cai, Reuben Tan, Jianrui Zhang, Bocheng Zou, Kai Zhang, Feng Yao, Fangrui Zhu, Jing Gu, Yiwu Zhong, Yuzhang Shang, et al. Temporalbench: Benchmarking fine-grained temporal understanding for multimodal video models. *arXiv preprint arXiv:2410.10818*, 2024. 2

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017. 6, 8, 1, 2

[8] Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jeng-Neng Hwang, Saining Xie, and Christopher D Manning. Auroracap: Efficient, performant video detailed captioning and a new benchmark. *arXiv preprint arXiv:2410.03051*, 2024. 2, 3

[9] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 190–200, 2011. 2, 3

[10] Jr-Jen Chen, Yu-Chien Liao, Hsi-Che Lin, Yu-Chu Yu, Yen-Chun Chen, and Yu-Chiang Frank Wang. Rextime: A benchmark suite for reasoning-across-time in videos. *arXiv preprint arXiv:2406.19392*, 2024. 3

[11] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Bin Lin, Zhenyu Tang, et al. Sharegpt4video: Improving video understanding and generation with better captions. *arXiv preprint arXiv:2406.04325*, 2024. 2, 3, 4

[12] Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13320–13331, 2024. 2

[13] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 2

[14] 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. 5, 1, 3

[15] Lisa Dunlap, Yuhui Zhang, Xiaohan Wang, Ruiqi Zhong, Trevor Darrell, Jacob Steinhardt, Joseph E Gonzalez, and Serena Yeung-Levy. Describing differences in image sets with natural language. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24199–24208, 2024. 3

[16] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 203–213, 2020. 8

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6202–6211, 2019. 8

[18] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. Agqa: A benchmark for compositional spatio-temporal reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11287–11297, 2021. 3

[19] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14375–14385, 2024. 3

[20] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad: Movie description in context. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18930–18940, 2023. 3

[21] Tengda Han, Max Bain, Arsha Nagrani, Gul Varol, Weidi Xie, and Andrew Zisserman. Autoad ii: The sequel-who, when, and what in movie audio description. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13645–13655, 2023.

[22] Tengda Han, Max Bain, Arsha Nagrani, Gül Varol, Weidi Xie, and Andrew Zisserman. Autoad iii: The prequel-back to the pixels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18164–18174, 2024. 3

[23] MD Zakir Hossain, Ferdous Sohel, Mohd Fairuz Shiratuddin, and Hamid Laga. A comprehensive survey of deep learning for image captioning. *ACM Computing Surveys (CsUR)*, 51(6):1–36, 2019. 1

[24] Hang Hua, Yunlong Tang, Chenliang Xu, and Jiebo Luo. V2xum-llm: Cross-modal video summarization

with temporal prompt instruction tuning. *arXiv preprint arXiv:2404.12353*, 2024. 7, 8, 9

[25] Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. Visual storytelling. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 1233–1239, 2016. 3

[26] 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)*, pages 18198–18208, 2024. 3

[27] Harsh Jhamtani and Taylor Berg-Kirkpatrick. Learning to describe differences between pairs of similar images. *arXiv preprint arXiv:1808.10584*, 2018. 3

[28] Qirui Jiao, Daoyuan Chen, Yilun Huang, Yaliang Li, and Ying Shen. Img-diff: Contrastive data synthesis for multimodal large language models. *arXiv preprint arXiv:2408.04594*, 2024. 3

[29] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6232–6242, 2019. 7

[30] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 706–715, 2017. 3

[31] Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. What matters when building vision-language models? *arXiv preprint arXiv:2405.02246*, 2024. 6

[32] Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *arXiv preprint arXiv:2206.03428*, 2022. 6, 1

[33] 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, 4, 6, 7, 8, 1

[34] Boyi Li, Ligeng Zhu, Ran Tian, Shuhan Tan, Yuxiao Chen, Yao Lu, Yin Cui, Sushant Veer, Max Ehrlich, Jonah Philion, et al. Wolf: Captioning everything with a world summarization framework. *arXiv preprint arXiv:2407.18908*, 2024. 3, 4

[35] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 1

[36] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Video storytelling: Textual summaries for events. *IEEE Transactions on Multimedia*, 22(2):554–565, 2019. 3

[37] 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 (ICML)*, pages 19730–19742. PMLR, 2023. 1, 2, 3

[38] Sheng Li, Zhiqiang Tao, Kang Li, and Yun Fu. Visual to text: Survey of image and video captioning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(4):297–312, 2019. 1

[39] Shicheng Li, Lei Li, Shuhuai Ren, Yuanxin Liu, Yi Liu, Rundong Gao, Xu Sun, and Lu Hou. Vitatecs: A diagnostic dataset for temporal concept understanding of video-language models. *arXiv preprint arXiv:2311.17404*, 2023. 3

[40] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26689–26699, 2024. 2, 3, 6, 1

[41] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26296–26306, 2024.

[42] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. 2, 3, 4

[43] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? *arXiv preprint arXiv:2403.00476*, 2024. 3

[44] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:46212–46244, 2023. 3

[45] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15579–15591, 2023. 8

[46] Asmar Nadeem, Faegheh Sardari, Robert Dawes, Syed Sameed Husain, Adrian Hilton, and Armin Mustafa. Narrativebridge: Enhancing video captioning with causal-temporal narrative. *arXiv preprint arXiv:2406.06499*, 2024. 3

[47] Nguyen Nguyen, Jing Bi, Ali Vosoughi, Yapeng Tian, Pooyan Fazli, and Chenliang Xu. Oscar: Object state captioning and state change representation. *arXiv preprint arXiv:2402.17128*, 2024. 3

[48] Dong Huk Park, Trevor Darrell, and Anna Rohrbach. Robust change captioning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4624–4633, 2019. 3

[49] Toby Perrett, Tengda Han, Dima Damen, and Andrew Zisserman. It's just another day: Unique video captioning by discriminative prompting. *arXiv preprint arXiv:2410.11702*, 2024. 3

[50] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024. 1

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learn-

ing transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021. 2

[52] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems (NeurIPS)*, 36, 2024. 3, 5

[53] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 2, 4, 5, 6, 1, 3

[54] 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)*, pages 14313–14323, 2024. 3

[55] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987. 2

[56] Ragav Sachdeva and Andrew Zisserman. The change you want to see (now in 3d). In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2060–2069, 2023. 3

[57] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only time can tell: Discovering temporal data for temporal modeling. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 535–544, 2021. 8

[58] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 1

[59] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1207–1216, 2019. 3, 5, 1, 2

[60] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):652–663, 2016. 2

[61] Jiawei Wang, Liping Yuan, and Yuchen Zhang. Tarsier: Recipes for training and evaluating large video description models. *arXiv preprint arXiv:2407.00634*, 2024. 2, 3, 4

[62] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 2, 3, 6, 7, 1

[63] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4581–4591, 2019. 2, 3

[64] 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. 8

[65] Yuxuan Wang, Yueqian Wang, Dongyan Zhao, Cihang Xie, and Zilong Zheng. Videohallucer: Evaluating intrinsic and extrinsic hallucinations in large video-language models. *arXiv preprint arXiv:2406.16338*, 2024. 3

[66] Ziyang Wang, Yi-Lin Sung, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Unified coarse-to-fine alignment for video-text retrieval. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2816–2827, 2023. 1

[67] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10704–10713, 2023.

[68] Wenhao Wu, Xiaohan Wang, Haipeng Luo, Jingdong Wang, Yi Yang, and Wanli Ouyang. Cap4video++: Enhancing video understanding with auxiliary captions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[69] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786, 2021. 3, 8

[70] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016. 2, 3

[71] 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. 4

[72] Zihui Xue, Kumar Ashutosh, and Kristen Grauman. Learning object state changes in videos: An open-world perspective. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18493–18503, 2024. 3, 5, 8, 1, 2

[73] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1686–1697, 2021. 1

[74] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10714–10726, 2023. 3

[75] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi,

Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 2, 3

[76] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4651–4659, 2016. 2

[77] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 192–199, 2014. 3

[78] Kuo-Hao Zeng, Tseng-Hung Chen, Ching-Yao Chuang, Yuan-Hong Liao, Juan Carlos Niebles, and Min Sun. Leveraging video descriptions to learn video question answering. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017. 1

[79] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986, 2023. 7

[80] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 2, 3

[81] Weiyu Zhang, Menglong Zhu, and Konstantinos G Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2248–2255, 2013. 6, 8, 1, 2

[82] 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, 3, 6, 1

[83] Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. Beyond hallucinations: Enhancing lvlms through hallucination-aware direct preference optimization. *arXiv preprint arXiv:2311.16839*, 2023. 3

[84] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018. 1

[85] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018. 2, 3

[86] 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)*, pages 18243–18252, 2024. 3

Progress-Aware Video Frame Captioning

Supplementary Material

1. Dataset

1.1. FrameCap Training Data

To construct the FramePair dataset, we employ a suite of open-source VLMs as captioners for initial pseudo label generation, including VILA [40], Qwen2-VL [62], LLAVA-Next-Video [35], LLAVA-Video [82] and LLAVA-OV [33]. Training videos are sourced from HowToChange and COIN, with frames extracted at 1FPS. We prepare pairs of frames for stage-I and multi-frame sequences for stage-II; the frame sequence length ranges from 3 to 6, as our preliminary experiments suggest that extending beyond 6 frames causes multiple issues with our captioners, such as overly brief captions, memory overflows, and great temporal mismatches.

We then process the data through our custom-designed tasks: progression detection and caption matching, to filter for high-quality data. The progression detection uses LLAMA-3.1-70B-Instruct [14], and for caption matching, we use VILA [40], chosen for its open-source availability and strong performance. Specifically, we assess caption matching precision by comparing model-generated answers against human responses on a subset of 90 questions. Gemini-1.5-Pro [53] achieves a precision of 0.89, while VILA achieves 0.75, the highest among open-source VLMs. Given that Gemini-1.5-Pro API usage incurs a cost, we reserve it for evaluation while utilizing the cost-free VILA as the caption matching evaluation VLM during the pseudo labeling stage.

For each frame sequence, the caption sequence that passes and fails these evaluations forms our preference data, which is utilized for DPO training of ProgressCaptioner. See Figure 10 and 11 for examples of frame pair data obtained from progression detection and caption matching, respectively, and Figure 12 for an illustration of the frame sequence data preparation process. Table 3 provides a summary of the training data statistics. The first data preparation stage collects a total of 240K frame-caption pairs for supervised fine-tuning (SFT) and 21K preference pairs for direct preference optimization (DPO). The second stage further expands the dataset to include 34K multi-frame and caption sequences for SFT, along with 26K frame-caption sequences for DPO.

1.2. FrameCapEval Benchmark

For the FrameCapEval benchmark, we source videos from four action-focused datasets: HowToChange [72], COIN [59], Penn Action [81] and Kinetics [7]. We ensure a balanced selection of videos from each action category

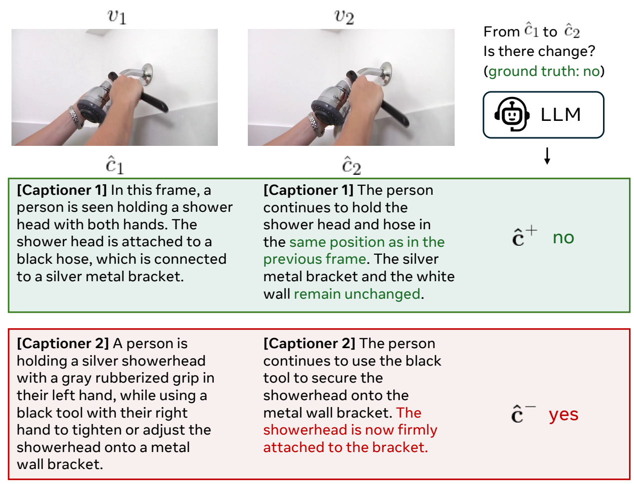


Figure 10. Example of a frame pair (decided by progression detection). The upper caption pair is marked as "accepted" by the evaluation LLM, aligning with the ground truth progression label (no progression), while the lower caption pair is marked as "rejected" because it incorrectly suggests progression.

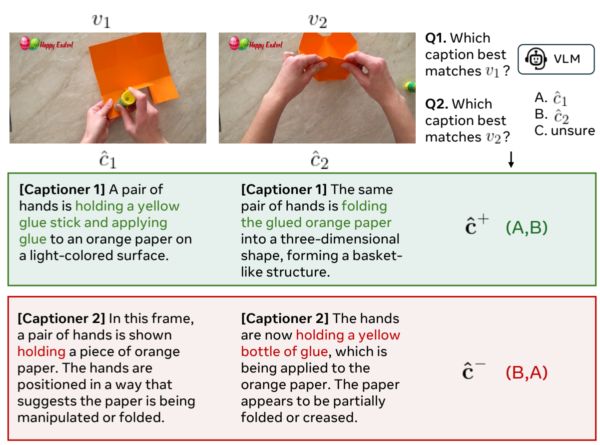


Figure 11. Example of a frame pair (decided by caption matching). The upper caption pair is marked as "accepted" since the evaluation VLM correctly answers the caption matching questions as (A, B), demonstrating good alignment. In contrast, the lower pair is "rejected" due to its answers (B, A), indicating poor correspondence between the frame and the generated captions.

across these datasets and follow their original validation or test splits. We are mindful of the single frame bias [32]—a recognized issue in video understanding where some actions are not distinctly temporal and can be adequately depicted with a single frame. To address this, we conduct a manual verification of all videos to eliminate frame sequences that lack fine-grained action progression, as these scenarios are straightforward and can be adequately man-

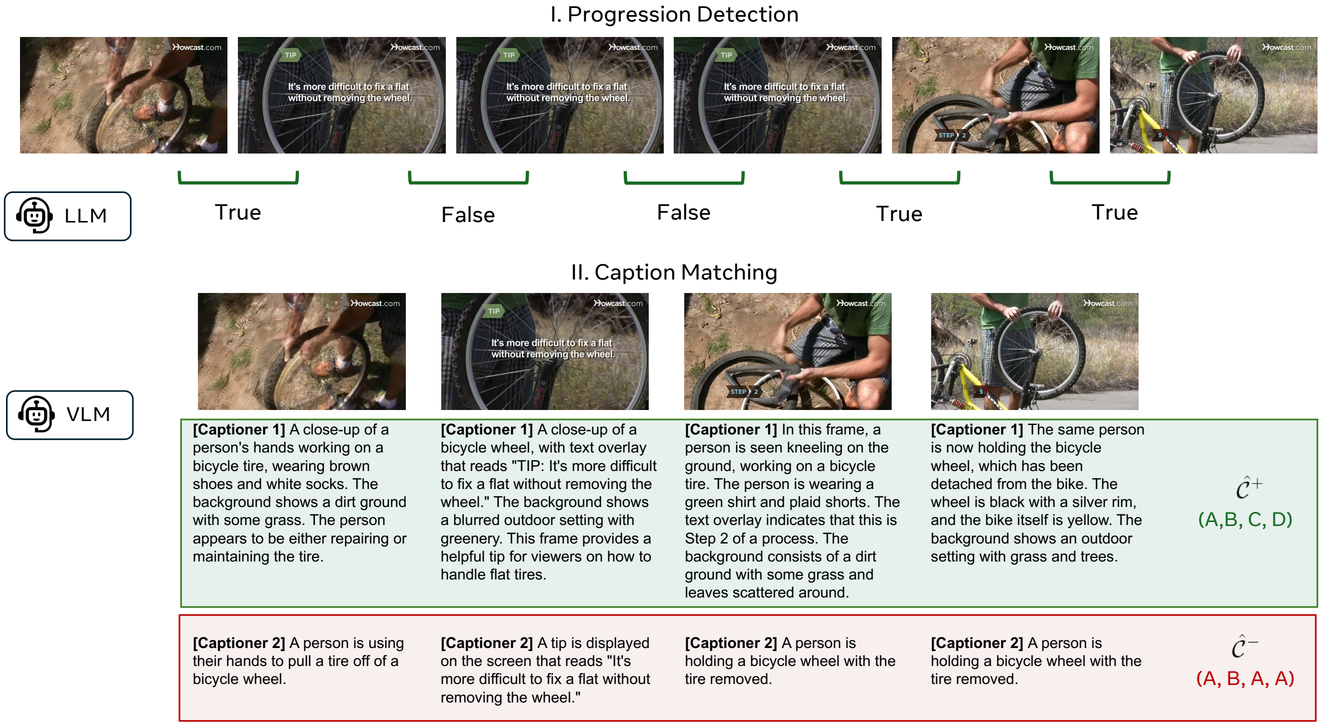


Figure 12. Example of a frame sequence. Progression detection is first applied to each adjacent frame pair to determine the visual-change label and identify M distinct frames. Caption matching then evaluates the captions corresponding to these M frames. The upper caption sequence is "accepted" as the evaluation VLM correctly answers (A, B, C, D), whereas the lower caption sequence, leading to erroneous responses, is marked as "rejected".

Dataset	# Videos	# Frames	# Pair		# Seq	
			SFT	DPO	SFT	DPO
HowToChange [72]	7,812	101,369	83,383	8,453	13,602	8,362
COIN [59]	9,030	103,791	156,858	12,622	20,704	17,826
Total	16,842	205,160	240,241	21,075	34,306	26,188

Table 3. We propose the FrameCap data collection, offering large-scale frame and caption sequences for fine-grained frame-level video captioning.

aged by image captioning models. Frames are extracted at 1 FPS and grouped using K-means clustering based on CLIP features [51], with K determined by silhouette scores [55] and ranging from 3 to 6. To each sequence, we add a frame with the smallest CLIP feature distance from a randomly chosen frame, so that the final sequence captures scenarios with and without action progression. See Table 4 for detailed evaluation data statistics.

FrameCap and FrameCapEval offer unique resources for temporally fine-grained descriptions at the frame level, which can be a valuable enhancement to current VLM's training data. We will publicly release the two datasets and hope that these resources help advance the temporal precision in video understanding capabilities of VLMs.

Dataset	# Videos	# Frames
HowToChange [72]	306 (102)	1101
COIN [59]	271 (139)	1063
Penn Action [81]	51 (47)	235
Kinetics600 [7]	56 (52)	451

Table 4. FrameCapEval data statistics. The numbers in parentheses represent the count of videos used for caption matching. We manually verify all selected frame sequences to assign action progression labels and filter out low-quality (easy) examples lacking clear action progression. This process ensures a robust testbed for evaluating a model's capability to generate temporally fine-grained descriptions.

2. Experiments

2.1. Experimental setup

Evaluation Metric Design Progression detection evaluates a model's action progress awareness, using caption pairs generated for each frame pair. It functions as a binary classification task, where label = 0 identifies scenarios with no visual progression to detect hallucinations, and label = 1 signifies visual progression to assess the model's ability to capture detailed temporal changes. We measure performance using *balanced accuracy*, which averages the true positive and true negative rates to account for data imbalance. To enhance the reliability and quality of our evaluations, we manually annotate visual progression between frames in the FrameCapEval dataset. Llama-3.1-70B-Instruct [14] is employed as the evaluation LLM to determine if a caption pair describes visual progression.

Caption matching assesses both the accuracy and the temporal granularity of captions. The evaluation is conducted on T-frame sequences that depict action progression, which are manually validated to ensure reliability. Gemini-1.5-Pro [53] is adopted as the evaluation VLM and tasked with performing the frame-wise caption matching task. We measure *sequence-level accuracy*, defined as the proportion of sequences where every frame is correctly identified by the evaluation VLM among all test sequences. It reflects how many caption sequences are entirely correct, which effectively rules out the possibility of random guessing being successful for a few frames within the sequence, providing a more robust assessment of caption sequence quality.

For the user study, 15 participants reviewed 85 frame sequences (364 frames) in total, randomly sampled from the FrameEvalBenchmark. We evaluate the captions from four leading models—two open-source (LLAVA-OV [33], Qwen2-VL [62]) and two proprietary (Gemini-1.5-Pro [53], GPT-4o [2])—alongside our ProgressCaptioner. Note that image captioning baselines are excluded due to their excessively lengthy captions and complete lack of temporal coherence. Participants are presented with captions produced by these five models, randomly shuffled for each sequence, and asked to choose the best and second best (with an additional "none" option available) for each frame's caption. The *average selection rate per model* is reported, providing insights into subjective caption quality preferences.

Implementation The Stage-I (frame pair captioning) and Stage-II (frame sequence captioning) models are trained with the same hyperparameters and undergo the same training processes: SFT followed by DPO. In the SFT phase, learning rates are set at 1e-5 for the LLM and projector, and 2e-6 for the vision encoder, with a batch size of 64. For DPO, the learning rate is reduced to 5e-7 with a batch size of 8. We set the preference scaling parameter α = 1.0 and the temperature parameter β = 0.2.

During inference, ProgressCaptioner takes frame sequences ranging from 2 to 6 frames. This limit is set because, as discussed earlier, all models experience severe performance degradation with longer frame sequences; hence, we cap at 6 frames when preparing training data and keep the inference protocol consistent with training. For sequences exceeding this length, ProgressCaptioner can operate in a sliding window mode.

For results in Section 4.1, direct inference is applied on T frames. For results in Section 4.2, we employ a 2-frame sliding window, where ProgressCaptioner performs frame pair captioning (except for NeXT-QA, where we uniformly sample 6 frames from the original video and apply direct inference on these 6-frame sequences without a sliding window). A single frame (v_t) can receive two captions: one from the pair (v_{t-1}, v_t) and another from (v_t, v_{t+1}). We concatenate the two captions for frame classification tasks to provide richer contextual information, aiding the LLM in frame label prediction. For keyframe selection, we use the caption from the pair (v_{t-1}, v_t) for v_t to maintain caption sequence coherence.

2.2. Prompt used

We design the following prompt for VLMs to perform the frame-wise video captioning task:

Caption Generation Prompt

Instructions:
These are `T` frames extracted from a video sequence depicting `action`. Provide a detailed description for each frame.

Requirement:
(1) Ensure each frame's description is specific to the corresponding frame, not referencing other frames.
(2) The description should focus on the specific action being performed, capturing the progression of the action. There is no need to comment on other elements, such as the background or unrelated objects.

Reply with the following format:
`<Frame 1>`: Your description
⋮
`<Frame T>`: Your description.

where `T` represents the number of frames in the sequence, and `action` is the video-level action label. The prompt is selected based on preliminary experiments on a small set of data, and we manually review the generated captions to ensure their effectiveness. We use the same prompt consistently for pseudo labeling training data and for evaluating current VLMs, both for our model and existing ones.

The progression detection prompt provided to the LLM is as follows:

Progression Detection Prompt (Pseudo-labeling)

Instructions:
You will be provided with two image descriptions. Your task is to determine the relationship between the two images based on these descriptions.

Image 1 description: `desc1`
Image 2 description: `desc2`

Choose the most appropriate option from the following:

A. The images likely look similar (no significant change).
B. There are noticeable changes between Image 1 and Image 2.
C. It is not possible to determine the similarity or difference based on the descriptions.

Progression Detection Prompt (Evaluation)

Instructions:
You will be provided with two image descriptions depicting an action. Your task is to determine the relationship between the actions in the two images based on the descriptions provided.

Action: `action`
The image descriptions are:
Image 1: `desc1`
Image 2: `desc2`

Choose one of the following options:

A. Action Progression: The action has advanced from Image 1 to Image 2 (e.g., more of the task has been completed in Image 2).
B. No Action Progression: The action remains the same between Image 1 and Image 2 (e.g., the images may show a change in viewpoint, hand position, or slight object adjustments, but the action itself has not progressed).
C. Uncertain: It is unclear whether the action has progressed or not.

In these prompts, `desc1` and `desc2` represent the descriptions of Image 1 and Image 2, respectively, and `action` is the video-level action label. The progression detection prompts differ between training and evaluation as they serve distinct purposes. For training, we aim to identify visually different frames within a sequence to ensure that the frame sequences processed later by caption matching are composed of distinct frames. Therefore, the training prompt focuses on detecting any visual changes, regardless of their nature. For evaluation, the objective shifts to determining whether the caption sequence is progress-aware; we manually annotate each frame sequence with progression labels for this purpose. As such, the evaluation prompt is designed to discern whether there is action progression or no action progression, rather than identifying simple visual changes. It is important to note that "changes" can encompass broader aspects than "progression", as explained in the prompt, changes may include viewpoint change or background object adjustments, which do not necessarily indicate a progression in the ongoing action.

Consider a sequence of M visually distinct frames $\mathcal{V}_M = \{v_i\}_{i=1}^M$, as detailed in Sec. 3.2. We task an evaluation VLM to perform caption matching for each frame $v_m \in \mathcal{V}_M$ with the following prompt:

Caption Matching Prompt

Which caption best describes the image?

<Frame v_m>

A. Caption $\hat{c}_1$

⋮

M. Caption $\hat{c}_M$
M+1. None of the above descriptions match the image, are hard to determine, or contain incorrect information about the image.

Reply with only the corresponding letter (A, B, C, etc.)

where <Frame v_m> denotes the image input (the m-th frame), and $\{\hat{c}_i\}_{i=1}^M$ is the caption sequence to be evaluated.

2.3. Results

Ablation Study Table 5 presents an ablation study using HowToChange videos from FrameCapEval, focusing on three key variables: (1) comparisons between Stage-I and Stage-II models; (2) the effect of training datasets—HowToChange alone versus HowToChange combined with COIN; (3) the impact of SFT alone versus SFT followed by DPO. The results demonstrate that all three factors are crucial for optimal performance. First, the Stage-I model, limited to frame pair captioning, does not provide caption matching accuracy for T-frame sequences and shows lower progression detection performance compared to the Stage-II model, which benefits from additional frame sequence training (see row 1 vs. row 2). Second, regarding training data, while evaluation is conducted on HowToChange, incorporating COIN data for training greatly improves performance, particularly in caption matching, highlighting the benefits of data scaling (see row 2 vs. row 4). This indicates potential for further enhancements by adding more datasets in the future. Finally, direct preference optimization (DPO) proves critical as its absence leads to great performance declines (see row 3 vs. row 4).

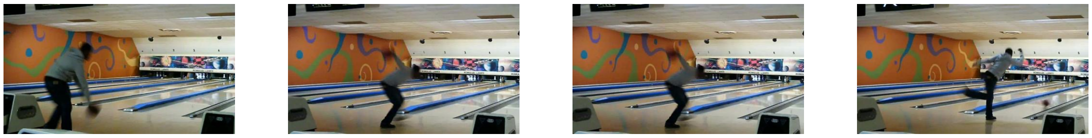
Model	Training Data	Training Strategy	Cap	Prog
Stage-I	HTC	SFT+DPO	–	70.6
Stage-II	HTC	SFT+DPO	28.4	73.1
Stage-II	HTC+COIN	SFT	24.5	68.3
Stage-II	HTC+COIN	SFT+DPO	**37.3**	**73.6**

Table 5. Ablation study of ProgressCaptioner on HowToChange (HTC) evaluation videos: examining the impact of model stages, training data, and optimization strategies on performance.

More Qualitative Results

Supplementing Figure 5 in the main paper, Figures 13–15 provide more qualitative predictions, comparing ProgressCaptioner with the four leading VLMs. As can be seen from these examples, while all baseline VLMs exhibit temporal inaccuracies in their descriptions, ProgressCaptioner consistently provides temporally precise and progress-aware captions, highlighting its superior performance.

Figure 13. Qualitative comparisons of ProgressCaptioner with SOTA VLMs (I). Red text identifies inaccuracies in the generated captions, while blue text highlights how our progress-aware captions build on prior content to clearly delineate what is changing or continuing.



[Gemini-1.5-Pro] The bowler is approaching the foul line, beginning their swing, with the ball held low.

[GPT-4o] The person is in the midst of a backswing, with their arm extended backwards, preparing to release the bowling ball.

[LLAVA-OV] A person is standing at the beginning of a bowling lane, preparing to bowl. They are wearing a gray hoodie and dark pants. The background features colorful abstract patterns on the wall.

[Qwen2-VL] A man is seen standing at the end of a bowling lane, preparing to throw a bowling ball.

[ProgressCaptioner] The person is standing at the start of a bowling lane, preparing to bowl. They are wearing a gray hoodie and dark pants, and there are several other lanes visible in the background with colorful patterns on the walls.

[Gemini-1.5-Pro] The bowler's arm is extended back in the backswing, preparing to bring the ball forward. Their body is leaning back slightly.

[GPT-4o] The person has fully raised their arm in the backswing, reaching the peak before beginning the forward motion to release the bowling ball.

[LLAVA-OV] The person has started their bowling motion, swinging their arm back in preparation to release the ball. Their body is slightly turned towards the lane, and their left leg is lifted behind them for balance.

[Qwen2-VL] The man has just thrown the bowling ball down the lane, and it is in motion.

[ProgressCaptioner] The person is in the process of delivering the ball down the lane. Their body is leaning forward, and their arms are extended behind them for balance.

[Gemini-1.5-Pro] The bowler is bringing their arm forward, starting the downswing, with the ball approaching the release point. Their body is starting to lean forward.

[GPT-4o] The individual is initiating the forward movement, transitioning from the backswing as they prepare to release the ball towards the lane.

[LLAVA-OV] The person is in the process of releasing the bowling ball down the lane. Their right arm is extended forward, and their body is leaning into the throw. The ball is just leaving their hand and starting its descent down the lane.

[Qwen2-VL] The bowling ball has reached the end of the lane and is about to hit the pins.

[ProgressCaptioner] The person is still in the process of delivering the ball down the lane. Their body is leaning forward, and their arms are extended behind them for balance.

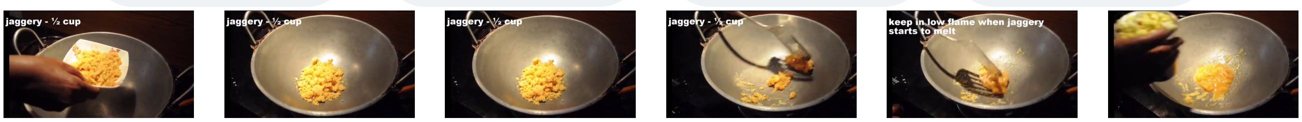
[Gemini-1.5-Pro] The bowler has just released the ball, their arm extended forward and down towards the lane. Their slide foot is near the foul line. The ball is visible rolling down the lane.

[GPT-4o] The person has released the bowling ball, with their arm extended forward and their stance indicating follow-through as the ball rolls towards the pins.

[LLAVA-OV] The person has completed their bowling motion, and the ball is now rolling down the lane. The person's right arm is still extended forward, and their body is following through with the throw. The ball is visible as it moves down the lane, heading towards the pins at the end.

[Qwen2-VL] The bowling ball has hit the pins, and the man is now standing at the end of the lane, likely preparing for his next turn.

[ProgressCaptioner] The person is in the follow-through phase of their delivery, with one leg lifted off the ground and their arms extended behind them for balance.



[Gemini-1.5-Pro] A hand is holding a white bowl filled with a pile of light brown jaggery over a metal pan.

[GPT-4o] A hand holds a bowl containing solid, crumbled jaggery about to be placed into a pan.

[LLAVA-OV] A hand is seen holding a small bowl containing jaggery, which is being poured into a large metal pan. The text overlay indicates that the quantity of jaggery being added is "1/2 cup".

[Qwen2-VL] A hand is seen holding a bowl of jaggery, which is a type of sugar made from sugarcane or sugar beets.

[ProgressCaptioner] A hand is holding a white bowl filled with yellow jaggery, which is being poured into a large metal pan. The text 'Jaggery - 1/2 cup' appears on the screen, indicating the quantity of jaggery being added.

[Gemini-1.5-Pro] The pile of light brown jaggery has been poured into the metal pan and sits at the bottom.

[GPT-4o] The crumbled jaggery is sitting in the center of the pan, still solid and unchanged from its previous state.

[LLAVA-OV] The jaggery has been fully poured into the pan and is now in the process of melting. The texture appears to be crumbly and granular.

[Qwen2-VL] The hand is now placing the jaggery into a pan on the stove.

[ProgressCaptioner] The jaggery has been poured into the metal pan, forming a small pile in the center. The text 'Jaggery - 1/2 cup' remains on the screen, emphasizing the amount of jaggery used.

[Gemini-1.5-Pro] The pile of light brown jaggery sits at the bottom of the metal pan, untouched.

[GPT-4o] The jaggery remains in the pan unchanged, retaining its solid, crumbled form.

[LLAVA-OV] The jaggery is being stirred with a utensil, likely a spatula or spoon, to ensure even melting. The color of the jaggery is a light golden brown.

[Qwen2-VL] The jaggery is now in the pan, and the hand is moving away from it.

[ProgressCaptioner] The image is identical to the previous frame, showing the same metal pan with a small amount of yellowish substance inside. There are no visible changes or actions taking place between the two frames.

[Gemini-1.5-Pro] A metal spoon is scooping up a spoonful of melting, sticky jaggery from the bottom of the metal pan.

[GPT-4o] A spatula is starting to move the jaggery, which begins breaking apart as it heats.

[LLAVA-OV] The stirring continues, and the jaggery is starting to melt and turn into a more fluid consistency. The pan's surface shows some residue from the melting process.

[Qwen2-VL] The hand is stirring the jaggery in the pan with a spoon.

[ProgressCaptioner] In this frame, a spoon is being used to stir the jaggery in the pan. The jaggery appears to be melting, and the spoon is actively mixing it. The text "Jaggery - 1/4 cup" is still present at the top of the image.

[Gemini-1.5-Pro] A metal spoon is lifting up a spoonful of melted, sticky jaggery from the bottom of the metal pan.

[GPT-4o] The spatula is actively stirring the jaggery, which is partially melted and turning into a thicker, sticky consistency.

[LLAVA-OV] The jaggery has almost completely melted, with only a few granules remaining. The liquid is a rich golden color, indicating it is ready for use in cooking.

[Qwen2-VL] The jaggery is now being melted in the pan, and the hand is stirring it continuously.

[ProgressCaptioner] The scene continues with the same metal pan on the stove, where the jaggery is now partially melted. A black spatula is seen stirring the jaggery, ensuring it melts evenly. Text appears at the bottom of the frame, reading "Keep in low flame when jaggery starts to melt," providing a cooking tip for the viewer.

[Gemini-1.5-Pro] A hand is holding a white bowl with some remaining jaggery over the metal pan, where the rest of the jaggery has melted into a light brown liquid.

[GPT-4o] The jaggery has mostly melted into a smooth liquid pool in the pan.

[LLAVA-OV] The jaggery has finished melting, leaving a smooth, golden liquid in the pan. The stirring utensil is still visible, suggesting the process was recently completed.

[Qwen2-VL] The jaggery is fully melted in the pan, and the hand is stirring it with a spoon.

[ProgressCaptioner] A hand holding a piece of food, possibly a vegetable or fruit, adds it to the pan with the melted jaggery. The focus is on the action of adding the ingredient to the pan, suggesting that the next step in the recipe involves combining the jaggery with other ingredients.

Figure 14. Qualitative comparisons of ProgressCaptioner with SOTA VLMs (II). Red text identifies inaccuracies in the generated captions, while blue text highlights how our progress-aware captions build on prior content to clearly delineate what is changing or continuing.

[Gemini-1.5-Pro] A brush is dipped into a black ink pool within a circular dish.

[GPT-4o] A hand is placing a small amount of water into a shallow dish, preparing the surface for ink.

[LLAVA-OV] A hand is seen holding a black ink pad over a white sheet of paper. The background shows a cluttered workspace with various art supplies scattered around, including paint tubes and brushes.

[Qwen2-VL] A person is seen holding a brush over a black ink container on a table.

[ProgressCaptioner] A close-up view of a black inkstone placed on a white sheet of paper, with various art supplies scattered around, including a red object and some colored pencils. A hand is seen holding the inkstone, indicating preparation for using sumi ink.

[Gemini-1.5-Pro] A rectangular ink stick is placed on the edge of the ink pool, partially submerged.

[GPT-4o] A hand is carefully positioning an ink stick vertically onto the dish, ready to begin grinding.

[LLAVA-OV] The hand places the ink pad down on the paper, creating a small puddle of black ink in the center. The workspace remains cluttered with art supplies visible in the background.

[Qwen2-VL] The person is now holding a wrapped piece of paper over the ink container.

[ProgressCaptioner] The same black inkstone is now shown with a small packet of sumi ink placed inside it. The background remains consistent with the previous frame, showing the same art supplies and a blue object partially visible on the left side.

[Gemini-1.5-Pro] The ink stick remains resting on the edge of the ink pool, partially submerged.

[GPT-4o] The ink stick is pressed further into the dish, indicating the start of the grinding process to create ink.

[LLAVA-OV] The hand picks up a small object, possibly a tool or a piece of equipment, from the cluttered workspace. The focus remains on the ink pad and paper, with the background still showing the scattered art supplies.

[Qwen2-VL] The person is still holding the wrapped paper over the ink container.

[ProgressCaptioner] The scene remains largely unchanged from the previous frame. The black inkstone is still on the white sheet of paper, and the small packet of sumi ink is still placed on top of the inkstone. There are no noticeable changes in the objects' positions or states between the two frames.

[Gemini-1.5-Pro] The ink stick is being moved within the ink pool, grinding against the bottom of the dish.

[GPT-4o] Both hands are holding the dish and the ink stick, actively grinding the ink stick in a circular motion to mix with the water.

[LLAVA-OV] The hand uses the small object to manipulate the ink on the paper, spreading it out slightly. The workspace remains cluttered, but the focus stays on the ink pad and paper.

[Qwen2-VL] The person is now holding the wrapped paper over the ink container and appears to be dipping the brush into the ink.

[ProgressCaptioner] In this frame, a pair of hands is visible, reaching towards the black inkstone that contains the sumi ink. The hands appear to be about to interact with the inkstone, possibly to mix or apply the ink. The background remains the same as in the previous frame, with the same scattered art supplies and stains on the surface.

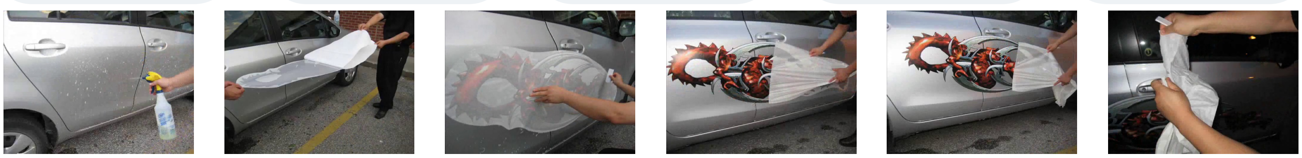
[Gemini-1.5-Pro] The ink stick continues to be ground against the bottom of the dish in a circular motion within the ink pool.

[GPT-4o] The grinding continues with increased vigor as the ink darkens and spreads throughout the water in the dish.

[LLAVA-OV] The hand continues to spread the ink on the paper using the small object. The workspace remains cluttered, but the focus remains on the ink pad and paper, with the ink starting to spread out more evenly across the surface.

[Qwen2-VL] The person is now holding the wrapped paper over the ink container and appears to be dipping the brush into the ink.

[ProgressCaptioner] The same person's hands are now dipping a black brush into the black inkstone, which contains a dark liquid, likely sumi ink. The background remains consistent with the previous frame, showing the same cluttered workspace with various art supplies.



[Gemini-1.5-Pro] Spraying a soapy solution onto the car door where the sticker will be applied.

[GPT-4o] A person is spraying a cleaning solution onto the side of a car.

[LLAVA-OV] A person is spraying a silver car with a yellow spray bottle, likely applying a cleaning solution or adhesive to the surface.

[Qwen2-VL] A person is seen holding a spray bottle and spraying a liquid onto a silver car.

[ProgressCaptioner] A close-up of a silver car door, where a hand is seen spraying water from a yellow and blue spray bottle onto the surface. The water droplets are visible on the car's paint, indicating that the car is being cleaned or prepared for further treatment.

[Gemini-1.5-Pro] Two people are carefully positioning a large sticker with a transparent backing onto the wet car door.

[GPT-4o] Two people are holding a large car sticker, preparing to apply it to the vehicle.

[LLAVA-OV] The same person is now holding a large white sheet of material, possibly a protective film or decal, and preparing to apply it to the car's door.

[Qwen2-VL] Two people are holding a white sheet and placing it on the car.

[ProgressCaptioner] Two individuals are standing next to the same silver car. One person is holding a large white sheet, which appears to be a protective cover or wrap for the car. The other person is holding the edge of the sheet, suggesting they are about to apply it to the car. The background shows a paved area with a yellow line, possibly indicating a parking spot or a boundary.

[Gemini-1.5-Pro] One person is using a squeegee to smooth out the sticker against the car door, working from the center outwards. The backing paper is still mostly covering the sticker.

[GPT-4o] A person is using a tool to smooth out the sticker on the car door.

[LLAVA-OV] The person is carefully positioning the white sheet onto the car's door, aligning it with the intended design area.

[Qwen2-VL] The person is seen peeling off the white sheet from the car.

[ProgressCaptioner] A close-up of a person's hands applying a red and black sticker with a dragon design onto the side of the same silver car. The person smooths out the sticker with their fingers, ensuring it adheres properly to the car's surface.

[Gemini-1.5-Pro] The backing paper is being peeled away from the car door, revealing a portion of the red and silver dragon design of the sticker.

[GPT-4o] A person is peeling back the backing paper from one edge of the sticker, revealing the design.

[LLAVA-OV] The person is smoothing out the white sheet on the car's door, ensuring there are no air bubbles or wrinkles.

[Qwen2-VL] The person is still peeling off the white sheet from the car.

[ProgressCaptioner] The same person is now peeling off a transparent plastic sheet from the car's surface, revealing the red and black dragon design underneath. The person's hands are visible as they carefully remove the sheet, exposing more of the intricate design.

[Gemini-1.5-Pro] Continuing to peel the backing paper off the sticker, revealing more of the dragon design. The sticker is now largely adhered to the car door.

[GPT-4o] The person continues to peel away more of the backing paper, further exposing the sticker's design.

[LLAVA-OV] The person is peeling off the protective backing from the white sheet, revealing the design underneath.

[Qwen2-VL] The person is seen peeling off the white sheet from the car.

[ProgressCaptioner] The same silver car with the red and black dragon design on its side. The person continues to peel off the white paper from the sticker, revealing more of the design. The background remains consistent with the paved surface and some dark spots.

[Gemini-1.5-Pro] Carefully peeling off the remaining backing paper from the sticker, revealing the complete dragon design. The sticker application is almost finished.

[GPT-4o] The person is holding and discarding the backing paper completely, having removed it from the sticker.

[LLAVA-OV] The final frame shows the person removing the last piece of the protective backing, exposing the fully applied design on the car's door.

[Qwen2-VL] The person is seen peeling off the white sheet from the car.

[ProgressCaptioner] The same person is now holding a small piece of white paper or cloth in their hands, examining it closely. The background shows the interior of the car, including the door handle and part of the dashboard.

Figure 15. Qualitative comparisons of ProgressCaptioner with SOTA VLMs (III). Red text identifies inaccuracies in the generated captions, while blue text highlights how our progress-aware captions build on prior content to clearly delineate what is changing or continuing.

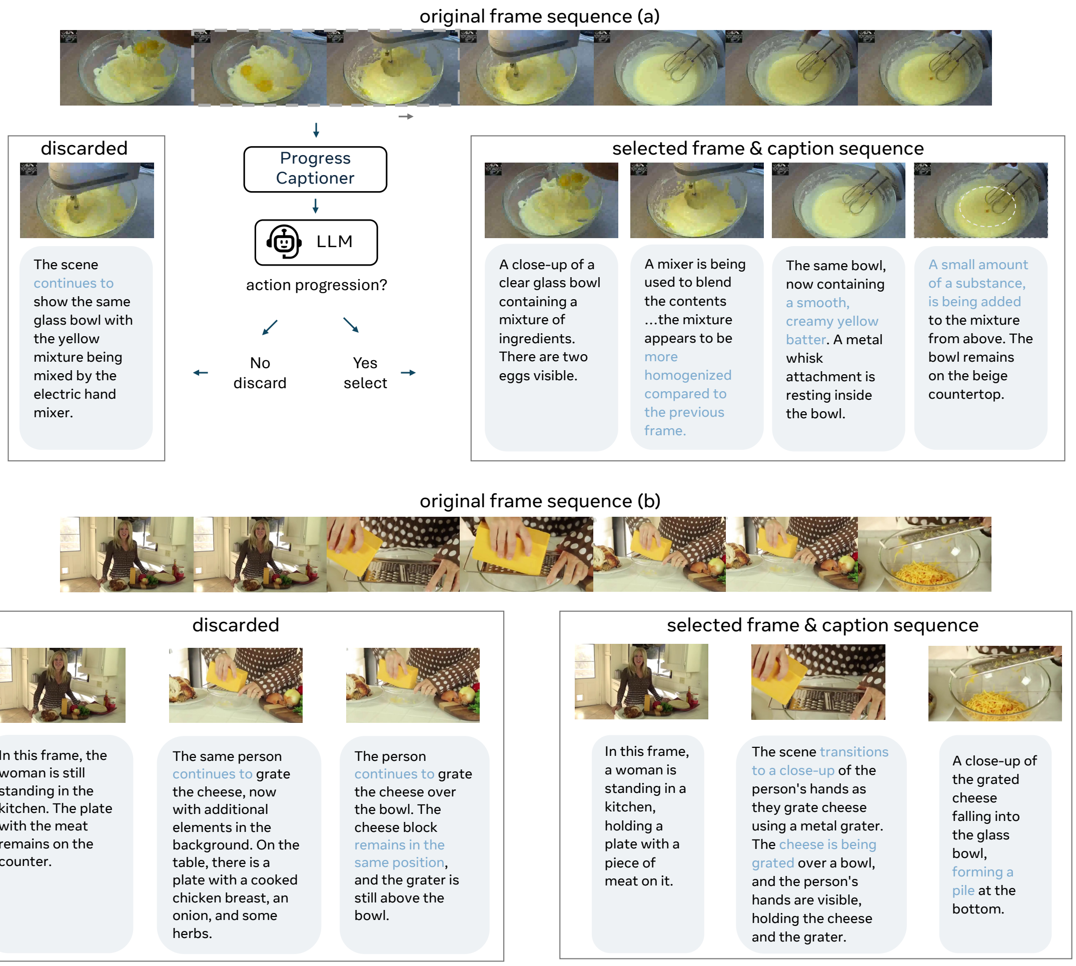


Figure 16. Captions produced by ProgressCaptioner and processed by an LLM enable us to automatically select representative frames that clearly depict action progression from densely sampled frame sequences. For each frame sequence, the bottom left box displays discarded frames alongside their captions, while the bottom right box showcases selected frames and their corresponding captions. This process effectively removes duplicate frames that depict the same action progression and enhances the selected frames with captions.

Keyframe Selection We propose to utilize frame-wise captions from ProgressCaptioner to select frames that depict action progression. The key idea is to "encode" a sequence of densely sampled video frames into per-frame captions, allowing an LLM to subsequently "decode" and identify key frames from this rich textual representation. The temporally fine-grained descriptions act as a condensed frame representation, focusing on action progression while remaining robust to visual disturbances such as changes in viewpoint or background objects. Figure 16 illustrates one potential design for such a keyframe selection feature. With ProgressCaptioner, we employ a sliding two-frame window for captioning, followed by an LLM (we use Llama-3.1-70B-Instruct) processing the generated captions. Specifically, for a sequence of densely sampled frames $\{v_t\}_{t=1}^{T}$, starting from $t = 1$, ProgressCaptioner generates caption (c_1, c_2) for (v_1, v_2). We then ask the LLM to determine if there is action progression between c_1 and c_2. If the answer is yes, frame v_2 gets selected; if no, v_2 is skipped to avoid redundancy as it likely depicts the same action stage as v_1. The process is repeated by advancing the window to (v_2, v_3) and continuing through the sequence.

Our approach offers two key advantages: (1) it efficiently filters out non-essential frames to ensure that selected frames distinctly represent action progression, and (2) it dynamically determines the size of the keyframe set based on the sequence content, eliminating the need for manually specifying the number of frames to sub-sample. To better illustrate this, we compare our method with the pseudo labeling strategy used in a recent video summarization work, V2Xum [24]. V2Xum employs an image captioning model followed by an LLM to perform extractive document summarization based on per-frame captions for keyframe selection.

Figure 17. Comparison of our keyframe selection with V2Xum [24]. Leveraging precise and progress-aware captions from ProgressCaptioner, our approach selects keyframes that accurately represent stages of the action process. In contrast, V2Xum's method often includes duplicate frames or overlooks frames that show subtle but important differences.

As shown in Figure 17, V2Xum's approach results in duplicate keyframes for sequence (a), where the first and second frames depict the same action progression despite a viewpoint change, and the last three frames similarly represent the action progression of oranges being sliced in half. In contrast, our method, leveraging the more accurate and temporally fine-grained captions produced by ProgressCaptioner, precisely identifies three distinct stages of this slicing action sequence. For sequence (c), V2Xum selects only one frame from the first four, despite depicting various stages of cutting a sausage (from whole to partially cut, fully cut, and then to chunks). Conversely, our approach accurately identifies all these frames as markers of action progression. It adaptively determines the size of the keyframe set, which can vary from small to large depending on the actual content, offering flexibility without requiring manual specification.

To conclude, our keyframe selection approach effectively highlights critical moments within action sequences. We believe such a system has significant potential for providing focused insights in educational tutorials and sports analysis, benefiting learners and analysts alike.

Limitations Despite the enhanced performance of ProgressCaptioner, it still faces several challenges. Firstly, while we have developed an advanced pseudo labeling refinement process, the training data sourced from existing VLMs inherently limits the quality of the captions. Moreover, the automation of data filtering using evaluation LLMs and VLMs introduces noise—though less costly, it's not as reliable as human annotation. Secondly, we observe that captioning longer frame sequences presents increased difficulties; for instance, accurately captioning six-frame sequences is notably more challenging than two-frame sequences. Addressing this challenge to extend ProgressCaptioner's capabilities to handle longer sequences remains a critical area for future development. In addition, Figure 18 illustrates a failure case where ProgressCaptioner produces identical captions for the last four frames, failing to recognize fine-grained spatial changes—an area that current VLMs consistently fall short of. This underscores the need for further advancements in this area.

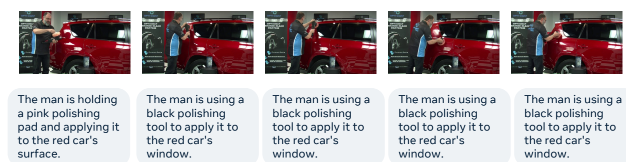


Figure 18. One failure case of ProgressCaptioner, where it fails to discern fine-grained spatial differences among the last four frames and thus produces identical captions.

Finally, we emphasize that the task of video frame captioning introduces a significant challenge by demanding high temporal precision. We recognize the limitations of ProgressCaptioner in its current stage and view this work as an initial step toward resolving this problem.