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 [83] 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 11 and 12 for examples of frame pair data obtained from progression detection and caption matching, respectively, and Figure 13 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 [82] and Kinetics [7]. We ensure a balanced selection of videos from each action category

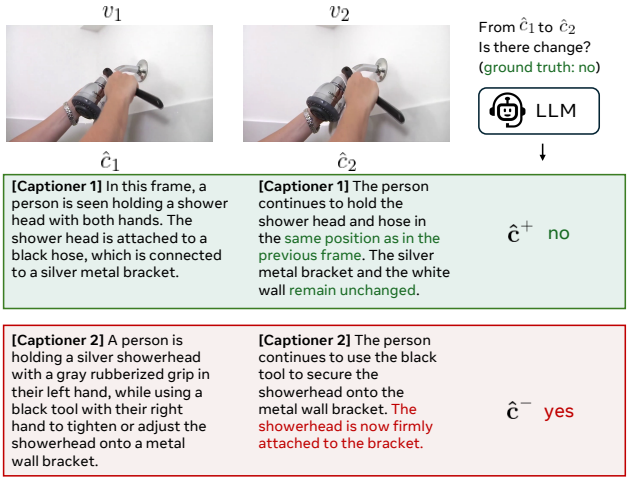


Figure 11. Example of a frame pair (decided by progression detection). The upper caption pair is marked as "accepted" by the evaluation LLM, aligning with the ground truth progression label (no progression), while the lower caption pair is marked as "rejected" because it incorrectly suggests progression.

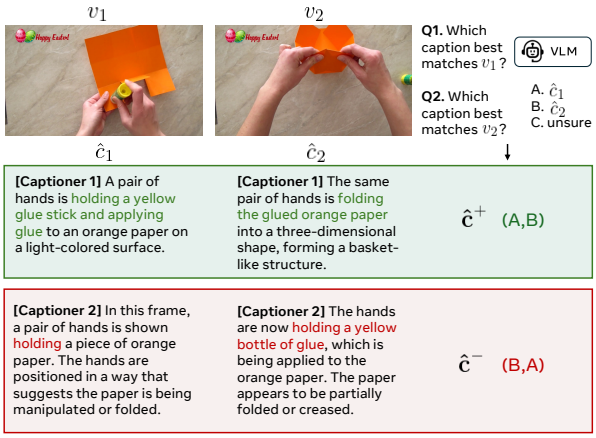


Figure 12. Example of a frame pair (decided by caption matching). The upper caption pair is marked as "accepted" since the evaluation VLM correctly answers the caption matching questions as (A, B), demonstrating good alignment. In contrast, the lower pair is "rejected" due to its answers (B, A), indicating 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-

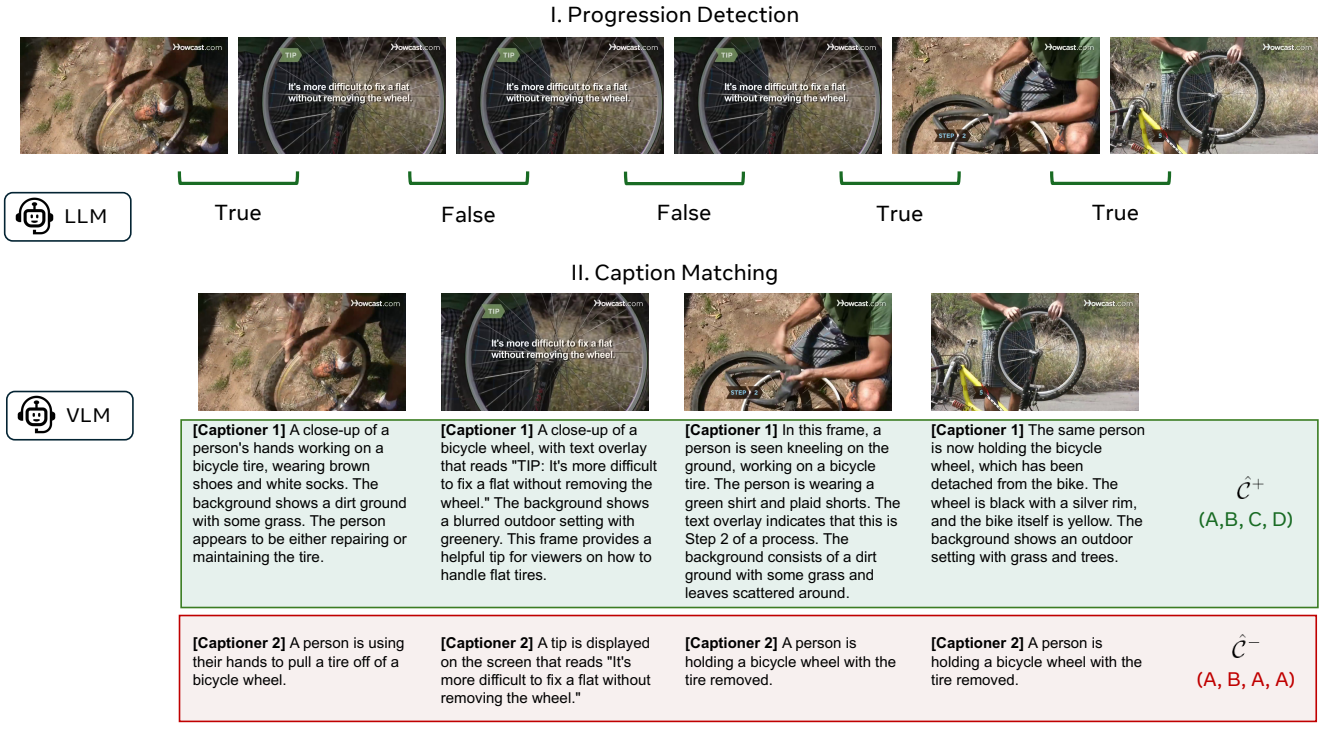


Figure 13. Example of a frame sequence. Progression detection is first applied to each adjacent frame pair to determine the visual-change label and identify M distinct frames. Caption matching then evaluates the captions corresponding to these M frames. The upper caption sequence is "accepted" as the 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.

Dataset	# Videos	# Frames
HowToChange [72]	306 (102)	1101
COIN [59]	271 (139)	1063
Penn Action [82]	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.

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.

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.

User Study To evaluate the subjective quality of generated captions, we conduct a user study involving 15 graduate student participants fluent in English. The study utilized 85 randomly sampled frame sequences (totaling 364 frames) from the FrameCapEval benchmark. 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 **??**, 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 6 in the main paper, Figures 14–16 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 14. 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.

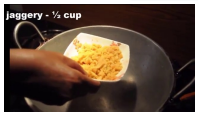
[Gemini-1.5-Pro] The bowler has just released the ball, their arm extended forward and down towards the lane. Their slide foot is near the foul line. The ball is visible rolling down the lane.

[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.



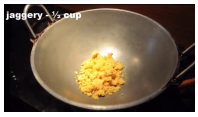
[Gemini-1.5-Pro] A hand is holding a white bowl filled with a pile of light brown jaggery over a metal pan.

[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.

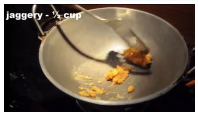
[Gemini-1.5-Pro] The pile of light brown jaggery sits at the bottom of the metal pan, untouched.

[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 15. 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.

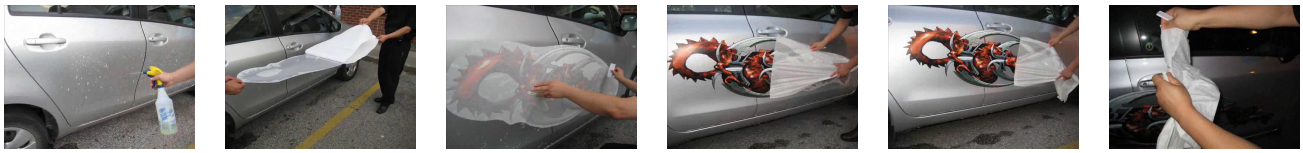
[Gemini-1.5-Pro] The ink stick continues to be ground against the bottom of the dish in a circular motion within the ink pool.

[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 16. 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.

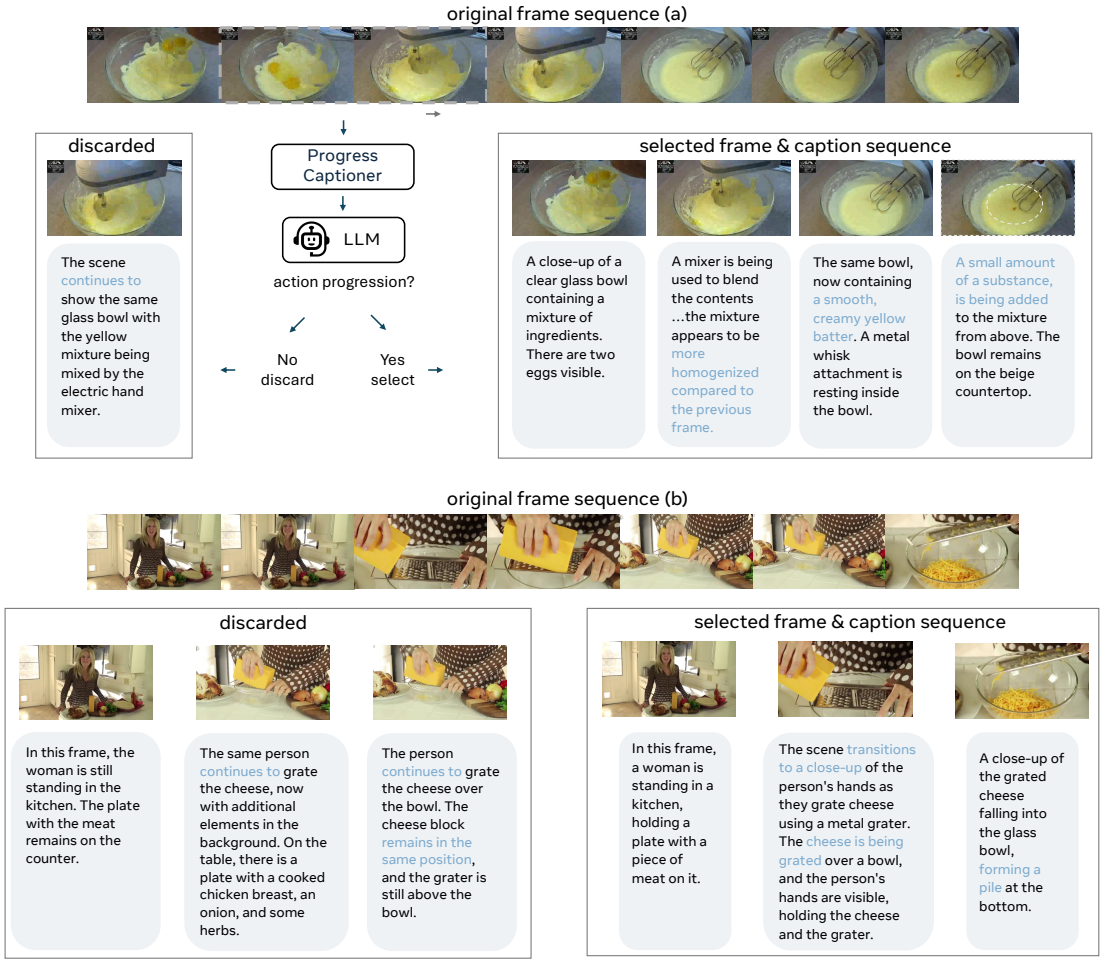


Figure 17. Captions produced by ProgressCaptioner and processed by an LLM enable us to automatically select representative frames that clearly depict action progression from densely sampled frame sequences. For each frame sequence, the bottom left box displays discarded frames alongside their captions, 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 17 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 18. 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 18, 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.

Justification of Automatic Evaluation Due to the lack of existing datasets with frame-wise ground truth captions, direct reference-based evaluation is infeasible. Therefore, we propose two automatic evaluation tasks, progression detection and caption matching, to assess frame-wise caption quality. To validate the reliability of these two metrics, we conduct experiments using different LLM/VLMs as evaluators for the two metrics (we pick the most widely adopted ones—Gemini and GPT for VLMs, Llama and GPT for LLMs). Figure 19 demonstrates consistent trends across these different evaluators, confirming the robustness of our evaluation methodology.

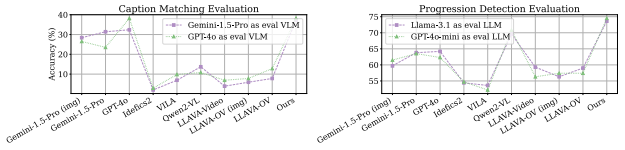


Figure 19. Caption matching (left) and progress detection (right) evaluation results on HowToChange, with different VLM/LLM as evaluators.

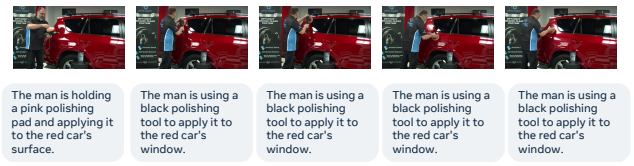


Figure 20. One failure case of ProgressCaptioner, where it fails to discern fine-grained spatial differences among the last four frames and thus produces identical captions.

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 20 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.

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.