Supplementary Materials: AXG-Reasoner: Error Detection and Explanation in Long Task Videos with Vision–Language Models

Shih-Po Lee
Northeastern University
lee.shih@northeastern.edu

Ehsan Elhamifar
Northeastern University
e.elhamifar@northeastern.edu

S.1. Detailed Prompts for LLM and VLM

We provide the full prompt templates for the LLM (Table S5) and VLM (Table S6). After subaction generation/summarization, we use the LLM to verify whether the subaction has been added to the action list ("Check if subaction exists" in Table S5) and whether it is a valid subaction for the corresponding action ("Check if subaction is valid" in Table S5), which helps filter out noisy or irrelevant subactions. For the VLM error reasoning prompts (P^c, P^r, and the Naive baseline), we include the task being performed ($< task\_name >$) and the steps/actions already completed ($< actions\_that\_have\_been\_done >$) to provide additional context.

S.2. Sinusoidal Embedding Function

We formulate the sinusoidal embedding described in Section 3.2 as follows:

$$\mathrm{PE}(t_{loc})_i = \begin{cases} \sin\left(t_{loc} \cdot 10000^{-2i/D}\right), & \text{if } i \text{ is even,} \\ \cos\left(t_{loc} \cdot 10000^{-2i/D}\right), & \text{if } i \text{ is odd.} \end{cases} \quad (1)$$

where $t_{loc} \in \{1, \ldots, 100\}$ is the time-stamp within an action segment, $\mathrm{PE}(t_{loc}) \in \mathbb{R}^D$, i is the dimension index, and D is the embedding dimension.

S.3. Efficiency in Error Reasoning

In this section, we evaluate both efficiency and efficacy of our AXG and Naive baseline during inference by examining performance with varying numbers of input keyframes (Table S1). We highlight two main observations. First, the Naive baseline steadily improves as more frames are provided, increasing from 51.9% with 3 frames to 59.7% with 12 frames on F1@.5. Second, AXG maintains stable performance across different numbers of frames for both error detection and explanation, indicating that AXG effectively selects informative keyframes through temporal subaction segmentation. Overall, AXG achieves higher performance (63.8% and 17.8% with 3 frames on F1@.5 and correctness score, respectively) than the Naive baseline (57.9% and 4.6% with 12 frames) and is more efficient, specifically when the number of subactions per action is four or fewer (see Figure S5).

Method	# of frames	Error Detection			Error Explanation
		N.	E.	F1	
Naive	3	84.0	19.8	51.9	1.0
Naive	8	87.1	26.3	56.7	3.6
Naive	12	**87.4**	**32.1**	**59.7**	**4.6**
AXG	3	79.7	**47.9**	**63.8**	17.8
AXG	8	80.1	47.0	63.6	17.4
AXG	12	**80.2**	47.0	63.6	**18.2**

Table S1. Error detection and explanation performance of different methods on EgoPER with GT action segments and Qwen2.5-VL as VLM.

Method	Temporal	Object	N.	E.	F1
AXG		✓	80.1	44.9	62.6
AXG	✓		79.3	43.9	61.6
AXG	✓	✓	**80.1**	**47.0**	**63.6**

Table S2. Ablation study of AXG for error detection on EgoPER using GT action segments. Temporal and Object denote temporally-aware textual embeddings and object names, respectively and Qwen2.5-VL as VLM.

S.4. Ablation Studies

Table S2 highlights the effectiveness of incorporating temporally aware textual embeddings for Kmeans clustering and leveraging object names during subaction generation. The results show that combining both components enables AXG to produce more accurate subactions and better localize their temporal positions, leading to improved error detection performance. In particular, AXG obtains its best performance with both components, achieving 63.6% on F1 and effectively detecting erroneous segments (47.0% on E.).

Furthermore, Table S3 presents qualitative comparisons of subactions generated with and without object names. In-

Action	AXG w/o Object	AXG
Check water temperature	1. inserting a thermometer into a kettle. 2. holding a thermometer.	1. inserting a thermometer into a kettle. 2. **checking the temperature of the water.**
Clean knife after spreading jelly	1. wiping the knife with a paper towel.	1. wiping the knife with a paper towel. 2. **tearing a piece of paper towel.**
Insert 5 toothpicks	1. picking up a toothpick.	**1. inserting a toothpick into the burrito.** 2. **picking up a toothpick from the container.**

Table S3. Qualitative examples for subactions generated by AXG w/o object names and our proposed AXG. Action "Check water temperature" is from task Tea in EgoPER. Action "Clean knife after spreading jelly" and "Insert 5 toothpicks into" are from task Pinwheel in EgoPER.

Method	EgoPER			Cook4D		
	BLEU-1	ROUGE-L	CIDEr	BLEU-1	ROUGE-L	CIDEr
Naive	0.1039	0.0566	0.0010	0.1511	0.0856	0.0060
AXG	**0.1783**	**0.1187**	**0.0472**	**0.2000**	**0.1801**	**0.0836**

Table S4. Average error explanation performance on BLEU, ROUGE-L, and CIDEr over tasks in EgoPER and CaptainCook4D. Naive and AXG use Qwen2.5-VL as VLM.

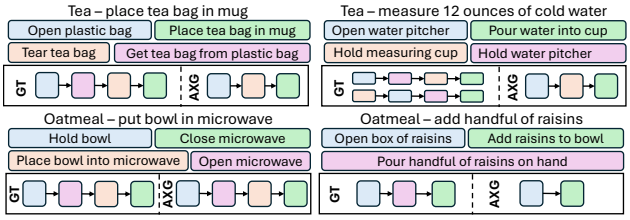

Figure S1. Qualitative results for GT and generated subactions.

corporating object names yields more specific and accurate descriptions (e.g., "checking the temperature of the water" vs. "holding a thermometer") and produces additional relevant subactions (e.g., "tearing a piece of paper towel" in "Clean knife after spreading jelly"), demonstrating the benefit of object-aware subaction generation.

S.5. Visualization for Subactions

In this section, we present qualitative results on subactions. We manually construct the ground-truth AXGs (GT AXGs) for four actions in EgoPER and provide qualitative comparisons in Figure S1 to demonstrate that our AXGs are semantically valid. However, GT AXGs may have annotator bias, as their structure can vary regarding to individual interpretations of subactions. Overall, our AXGs closely align with those constructed by human annotators. Figure S2, S3, and S4 show the illustration for temporal subaction segmentation generated by AXG for normal action segments. In summary, AXG effectively divides an action into subactions and localize their temporal posistions in an action segment. Specifically, AXG generates two subactions "opening a honey bottle" and "pouring honey into the mug" for action "add honey to mug" and four subactions "holding a bowl", "opening the microwave", "placing a bowl on the microwave", and "closing the microwave" for action "put bowl with water and oats in microwave".

S.6. Supplementary Experiments

We provide detailed performance on error detection (Table S7, S8, S9, and S10), the percentages of erroneous frames classified as actions or background by different TAS models (Table S11), and distribution of subaction counts per action under different clustering settings (Figure S5) for each task in EgoPER and CaptainCook4D.

In addition, Table S4 shows error explanation performance on BLEU-1, ROUGE-L, and CIDEr for EgoPER and CaptainCook4D. Overall, our AXG still outperforms the Naive baseline on all metrics. Note that the LLM generates free-form, semantically rich, and diverse text for error explanation, but classical metrics BLEU-1, ROUGE-L, and CIDEr for evaluating translation and captioning rely on n-gram overlap instead of semantic understanding. Valid predictions generated by the LLM may not share the same wording with the reference. These metrics cannot evaluate correctness or reasoning quality, and often produce near-zero scores even for accurate explanations. Therefore, they are unsuitable for assessing LLM-generated free-form text in error explanation and we instead rely on a LLM for assessment in our main paper.

S.7. More Qualitative Results

In this section, we present additional qualitative results for error explanation in Figures S6, S7, and S8. In Figure S6, the Naive baseline incorrectly interprets the user's actions, mistakenly claiming that the person picks up a tortilla from the floor and places it on the cutting board. In reality, the person discards the dropped tortilla and gets a new one from the package before placing it on the cutting board. In contrast, AXG correctly localizes the subaction "place tortilla on cutting board" and generates accurate explanations for both the dropped segment (e.g., "placing the tortilla on the floor instead of the cutting board") and the corresponding subaction segment (e.g., "clearly placing a tortilla on a

red cutting board"), effectively capturing the short temporal cues critical for error reasoning.

A similar case appears in Figure S8, where AXG detects the error "some grounds appear to have spilled onto the table" at the end of the action "pour coffee grounds into the filter cone". Finally, in Figure S7, the Naive baseline again fails to detect that the tortilla is placed on the table instead of the cutting board. AXG, however, produces a correct explanation for the dropped segment "the tortilla is not being placed on the cutting board; instead, it is left on the table.", capturing both spatial and temporal cues for error reasoning.

Overall, these results demonstrate that once erroneous segments are properly localized, providing explicit guidance (e.g., "you have made a mistake") within the VLM prompt significantly enhances the model's ability to explain errors clearly and precisely.

S.8. Limitations

In this section, we discuss several limitations related to the capabilities of current VLMs and the attributes of the dataset. First, Figure S9 shows that even when AXG correctly localizes a fine-grained subaction for error reasoning, the VLM may still fail to highlight sparse spatial cues. For example, it cannot recognize that the person is using a spoon instead of a knife. This suggests that VLMs still struggle to capture subtle visual differences, especially when small objects are involved.

Second, Figure S10 presents an error case in which "the person pours water into the wrong mug, which does not contain the tea bag already placed in another mug" during the action "pour water from kettle into mug". From the action description alone, this is not inherently an error, and the VLM provides a reasonable explanation. However, the EgoPER dataset considers it an error because the task cannot proceed until the tea bag is placed into the mug with water. This represents an implicit error that cannot be reliably inferred from a single segment and generic action description. Addressing such cases requires a more curated prompt or action description that incorporates relevant past context and the states of involved objects.

Finally, Figure S11 illustrates a mixed case of both issues. The VLM fails to identify banana slices as bananas in the action "put banana" and also overlooks that the person mistakenly places the banana slices into a bowl without oats which have been prepared in the other bowl.

Target	Prompt
Produce common object names	You are a system that given a list of object states, outputs at most five objects that most frequently appear. Select one of the output formats in the following: [object1], [object1];[object2], [object1];[object2];[object3], [object1];[object2];[object3];[object4], [object1];[object2];[object3];[object4];[object5]. List of object states: $< object\_states >$
Produce a common action as subaction	You are a system that given a list of actions, outputs only one sentence to describe the action that most commonly occurs. You do not need to output your reason. Select one of the output formats in the following: The person is [verb] [object1]. The person is [verb] [object1] [prep1] [object2]. The person is [verb] [object1] and [object2]. The person is [verb] [object1] [prep1] [object2] [prep2] [object3]. List of actions: $< list\_of\_actions >$
Check if subaction exists	You are a system that given a list of actions and a target action, answer the following question: does the target action highly match any action in the list and explain why? Focus the verb, objects, and preposition. Objects with different colors are considered matched. Output yes or no and the reason. The output format: Yes/No, Reason:... List of actions: $< existing\_subactions >$, Target action: $< subaction >$
Check if subaction is valid	You are a system that given a list of existing subactions and a new subaction, answer the following question: does the new subaction make any conflict with the existing subactions or the main action? The conflict indicates if the new subaction changes the objects used by the existing subactions or the object specified by the main action. Output yes or no and the reason. The output format: Yes/No, Reason:... Main action:$< action >$, List of subactions:$< existing\_subactions >$, New subaction:$< subaction >$
Error explanation evaluation	You are a system that given a reference description and a target description, outputs a score that represents the semantic similarity between two descriptions and your reason. Higher score means the two descriptions are more similar in semantics. The score is from 0 to 1. Output format: Score: x, Reason: y. Reference description: $< gt\_error\_description >$. Target description: $< predicted\_error\_explanation >$

Table S5. Detailed prompts for LLM.

Target	Prompt
P_{action}	You are a system that given a image, outputs the action the person is doing and the detailed state for every object the person is working with. The person is in the process of doing $< action >$. Do not use vague verbs such as prepare, manipulate, etc. If the person's hand is not visible, outputs 'the person is waiting.' Otherwise, select one of the output formats in the following to output the action: The person is [verb] [object1];The state of [object1] is [state1]. The person is [verb] [object1] [prep1] [object2];The state of [object1] is [state1];The state of [object2] is [state2]. The person is [verb] [object1] and [object2];The state of [object1] is [state1];The state of [object2] is [state2]. The person is [verb] [object1] [prep1] [object2] [prep2] [object3]. The state of [object1] is [state1];The state of [object2] is [state2];The state of [object3] is [state3].
P^c	You are trying to do $< task\_name >$. You have finished the following steps: $< actions\_that\_have\_been\_finished >$. Now you are trying to do an subaction $< subaction >$ of an action $< action >$. Given a sequence of images for $< subaction >$, output a score to show the correctness of the action and your reason. The score ranges from 0 to 1. Higher score indicates the action shown by those images is more correct. The output format: Score: x, Reason: y
P^r	You are trying to do $< task\_name >$. You have finished the following steps: $< actions\_that\_have\_been\_finised >$. Now you are trying to do an action $< action >$ and it seems like you are making a mistake. Given a sequence of images for $< action >$, use one sentence to describe what mistake you are making.
Naive baseline	You are trying to do $< task\_name >$. You have finished the following steps: $< actions\_that\_have\_been\_finished >$. Now you are trying to do an action $< action >$. Given a sequence of images for $< action >$, output a score to show the correctness of the action and your reason. The score ranges from 0 to 1. Higher score indicates the action shown by those images is more correct. The output format: Score: x, Reason: y

Table S6. Detailed prompts for VLM.

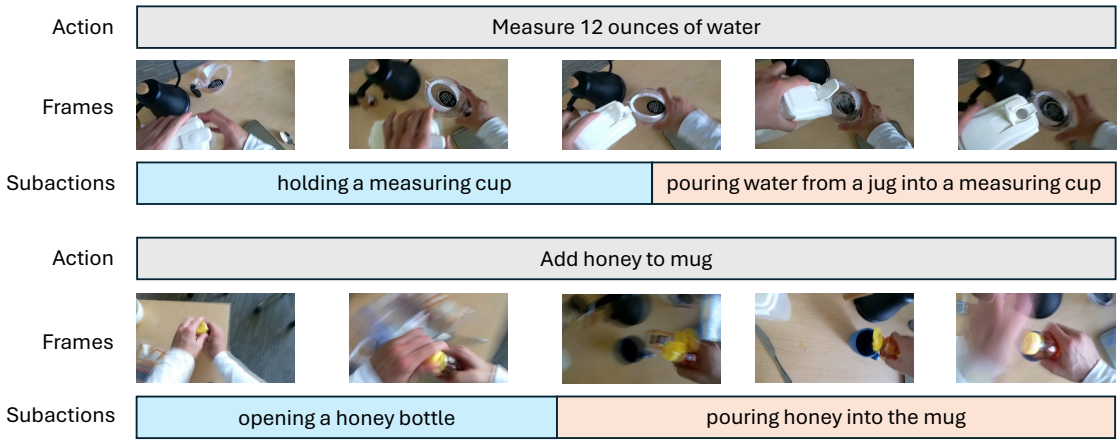

Figure S2. Visualization of temporal subaction segmentation for task Tea in EgoPER.

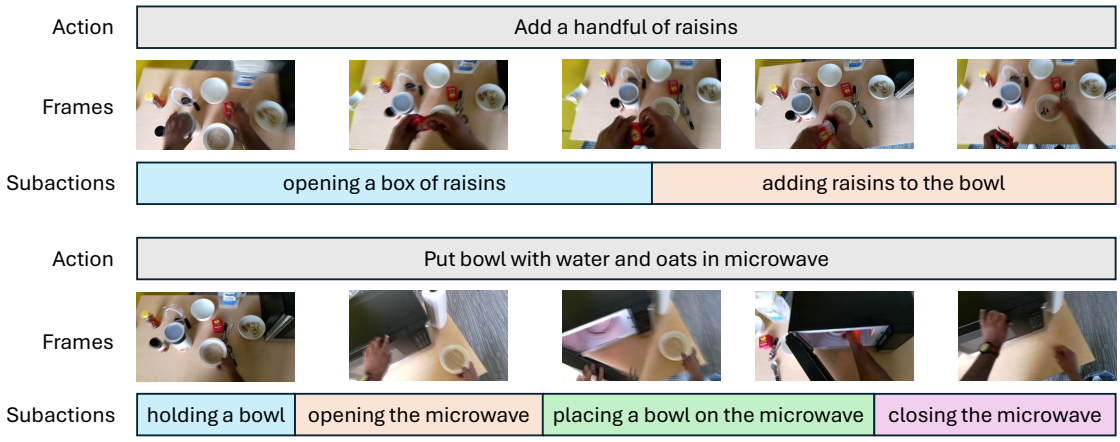

Figure S3. Visualization of temporal subaction segmentation for task Oatmeal in EgoPER.

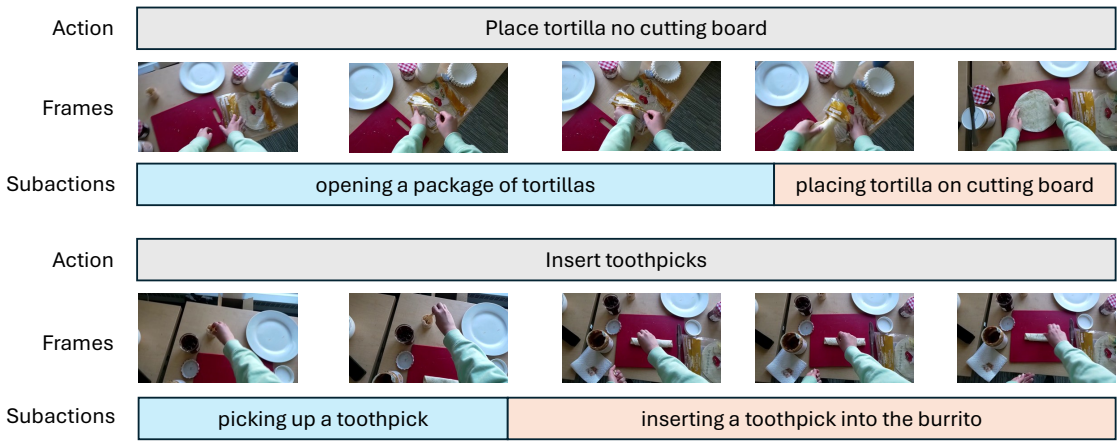

Figure S4. Visualization of temporal subaction segmentation for task Pinwheel in EgoPER.

Method	TAS	Quesadilla			Oatmeal			Pinwheel			Coffee			Tea			All		
		N.	E.	F1@.5	N.	E.	F1@.5	N.	E.	F1@.5	N.	E.	F1@.5	N.	E.	F1@.5	N.	E.	F1@.5
EgoPED	ACTF	30.1	25.2	27.7	29.3	12.9	21.1	14.6	24.5	19.6	9.5	8.9	9.2	36.3	34.0	35.1	24.0	21.1	22.5
GTG2Vid	GTG2Vid	40.4	22.6	31.5	58.5	36.4	47.4	30.6	16.0	23.3	33.1	4.0	18.6	51.5	37.0	44.2	42.8	23.2	33.0
Qwen2.5-VL-32B																			
Naive	ACTF	18.3	35.1	26.7	5.9	13.8	9.9	13.7	31.2	22.5	7.7	13.3	10.5	7.2	26.6	16.9	10.6	24.0	17.3
AXG	ACTF	34.1	28.1	31.1	33.1	20.9	27.0	17.3	20.8	19.1	20.8	9.6	15.2	41.7	39.7	40.7	27.4	24.4	25.9
Naive	GTG2Vid	33.9	11.9	22.9	32.2	13.5	22.8	27.2	9.1	18.2	43.7	5.3	24.5	15.8	11.2	13.5	30.6	10.2	20.4
AXG	GTG2Vid	45.0	35.1	40.0	39.4	30.7	35.0	26.6	28.5	27.6	30.8	17.3	24.0	40.4	40.5	40.4	36.4	30.4	33.4
InternVL3.5-14B																			
Naive	ACTF	31.4	9.8	20.6	45.2	10.8	28.0	28.7	11.2	19.9	22.5	1.9	12.2	26.9	17.4	22.2	31.0	10.2	20.6
AXG	ACTF	27.3	25.0	26.1	30.3	17.8	24.0	15.1	20.5	17.8	18.2	11.0	14.6	37.2	37.2	37.2	30.5	20.7	25.6
Naive	GTG2Vid	29.4	9.3	19.4	38.0	9.2	23.6	19.7	9.8	14.7	36.3	7.8	22.0	27.1	9.3	18.2	30.1	9.1	19.6
AXG	GTG2Vid	38.7	28.9	33.8	42.4	28.9	35.6	19.8	22.2	21.0	32.2	16.0	24.1	39.5	38.0	38.8	38.3	25.2	31.8

Table S7. Error detection performance of different methods on EgoPER.

Method	TAS	Hot Chocolate			Sandwich			Burritos			Ramen			Raita			All		
		N.	E.	F1@.5	N.	E.	F1@.5	N.	E.	F1@.5	N.	E.	F1@.5	N.	E.	F1@.5	N.	E.	F1@.5
EgoPED	ACTF	8.1	2.4	5.2	16.3	6.8	11.5	16.3	8.2	12.2	12.4	10.7	11.6	14.1	9.5	11.8	13.4	7.5	10.5
GTG2Vid	GTG2Vid	15.8	20.9	18.3	16.7	12.8	14.8	16.9	19.7	18.3	22.6	8.2	15.4	18.2	9.9	14.0	18.0	14.3	16.2
Qwen2.5-VL-32B																			
Naive	ACTF	13.8	23.2	18.5	14.1	23.6	18.8	30.8	7.8	19.3	23.4	20.6	22.0	19.8	29.9	24.8	20.4	21.0	20.7
AXG	ACTF	8.6	28.9	18.8	11.1	31.5	21.3	42.2	16.9	29.6	20.0	20.7	20.3	24.2	53.6	38.9	18.1	28.0	21.1
Naive	GTG2Vid	16.4	23.3	19.8	12.2	18.7	15.5	15.4	25.9	20.7	12.7	4.1	8.4	9.8	14.9	12.3	13.3	17.4	15.3
AXG	GTG2Vid	21.1	47.5	34.3	17.9	40.0	28.9	12.6	47.1	29.8	19.4	32.9	26.1	23.9	23.4	23.6	19.0	38.2	28.5
InternVL3.5-14B																			
Naive	ACTF	17.9	24.0	21.0	16.3	15.2	15.7	17.5	6.7	12.1	21.6	11.4	16.5	16.5	21.4	19.0	18.0	15.7	16.9
AXG	ACTF	8.7	32.0	20.3	11.0	31.2	21.1	40.4	16.9	28.7	21.9	21.1	21.5	25.6	54.9	40.2	18.7	28.8	23.8
Naive	GTG2Vid	15.4	25.5	20.5	24.1	27.3	25.7	5.7	24.1	14.9	18.2	7.7	12.9	16.3	22.5	19.4	15.9	21.4	18.7
AXG	GTG2Vid	21.1	47.5	34.3	19.5	41.7	30.6	12.6	47.1	29.8	22.0	33.8	27.9	23.9	21.3	22.6	19.0	39.1	29.0

Table S8. Error detection performance of different methods on CaptainCook4D.

Method	Quesadilla			Oatmeal			Pinwheel			Coffee			Tea			All		
	N.	E.	F1	N.	E.	F1	N.	E.	F1	N.	E.	F1	N.	E.	F1	N.	E.	F1
Qwen2.5-VL-32B																		
Naive	80.9	26.5	53.7	92.5	33.0	62.8	80.5	22.1	51.3	93.3	3.5	48.4	88.3	46.2	67.2	87.1	26.3	56.7
VTREE [53]	78.3	14.2	46.3	91.4	28.9	60.1	78.1	20.9	49.5	92.7	3.1	47.9	87.4	44.0	65.7	85.6	22.2	53.9
AXG	82.3	59.9	71.1	84.4	45.9	65.1	61.5	44.0	52.8	84.0	19.4	51.7	88.5	65.7	77.1	80.1	47.0	63.6
InternVL3.5-14B																		
Naive	89.4	13.4	51.4	91.5	7.6	49.5	83.6	11.5	47.6	91.6	13.7	52.7	84.7	22.4	53.5	88.2	13.7	50.9
AXG	81.7	59.3	70.5	84.1	43.3	63.7	62.0	40.4	51.2	82.6	18.6	50.6	85.7	58.7	72.2	79.2	44.1	61.6

Table S9. Error detection performance of different methods with GT action segments on EgoPER.

Method	Hot Chocolate			Sandwich			Burritos			Ramen			Raita			All		
	N.	E.	F1	N.	E.	F1	N.	E.	F1	N.	E.	F1	N.	E.	F1	N.	E.	F1
Qwen2.5-VL-32B																		
Naive	33.6	21.9	27.8	52.4	28.4	40.4	56.0	49.1	52.5	70.9	18.3	44.6	52.3	31.8	42.1	53.0	29.9	41.5
VTREE [53]	35.4	44.4	39.9	48.5	35.6	42.0	62.3	38.8	50.5	69.8	24.9	47.4	48.0	29.6	38.8	52.8	34.7	43.7
AXG	8.7	75.6	42.2	19.0	60.1	39.5	16.2	66.7	41.4	63.5	53.0	58.2	16.1	72.0	44.0	24.7	65.5	45.1
InternVL3.5-14B																		
Naive	53.3	31.2	42.2	68.1	35.1	51.6	72.7	6.3	39.5	71.6	22.2	46.9	72.0	16.3	44.2	67.5	22.2	44.9
AXG	8.7	76.6	42.7	19.0	61.1	40.0	61.3	28.7	45.0	67.2	56.3	61.8	12.7	67.3	40.0	33.8	58.0	45.9

Table S10. Error detection performance of different methods with GT action segments on CaptainCook4D.

Method	Quesadilla		Oatmeal		Pinwheel		Coffee		Tea	
	Action (%)	BG (%)	Action (%)	BG (%)	Action (%)	BG (%)	Action (%)	BG (%)	Action (%)	BG (%)
GTG2Vid	86	14	73	27	91	9	85	15	85	15
EgoPED	78	22	64	36	93	7	92	8	92	8
FACT	83	17	75	25	90	10	84	16	84	16

EgoPER

	Hot Chocolate		Sandwich		Burritos		Ramen		Raita	
	Action (%)	BG (%)	Action (%)	BG (%)	Action (%)	BG (%)	Action (%)	BG (%)	Action (%)	BG (%)
GTG2Vid	94	6	94	6	93	7	92	8	72	28
EgoPED	77	23	86	14	89	11	81	19	77	23
FACT	92	8	87	13	90	10	94	6	80	20

CaptainCook4D

Table S11. The percentages (%) of erroneous frames classified as actions or background class (BG) by different TAS models for each task in EgoPER and CaptainCook4D.

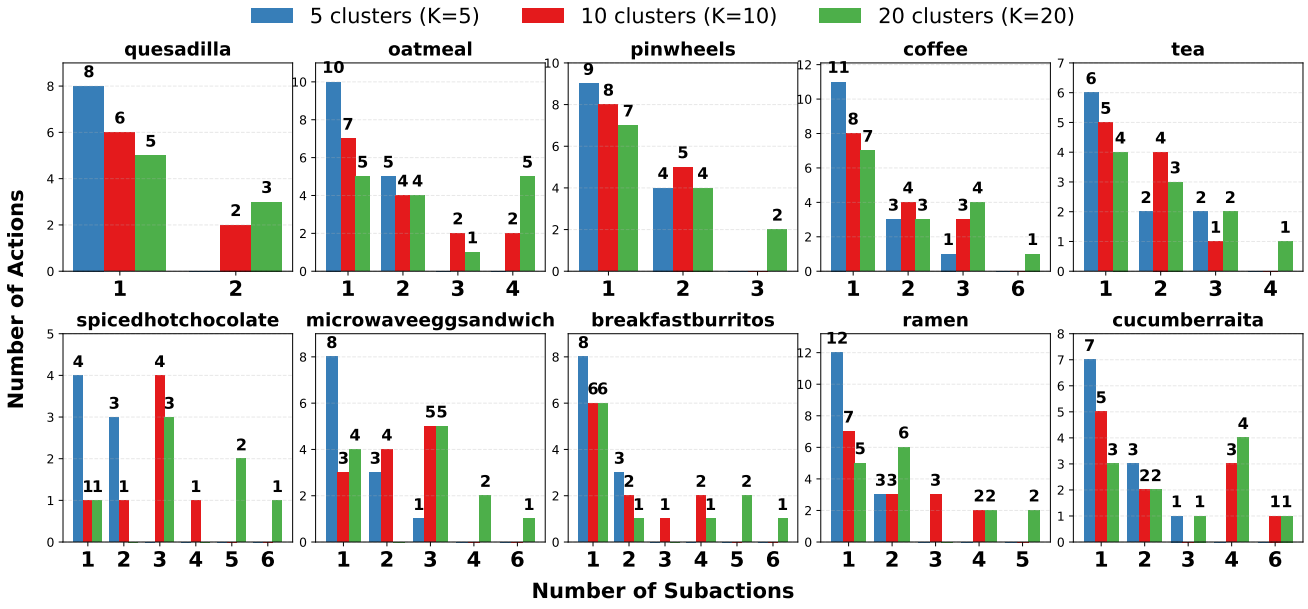

Figure S5. Distribution of the number of subactions per action for each task of EgoPER and CaptainCook4D. Each bar shows how many actions contain a given number of subactions..

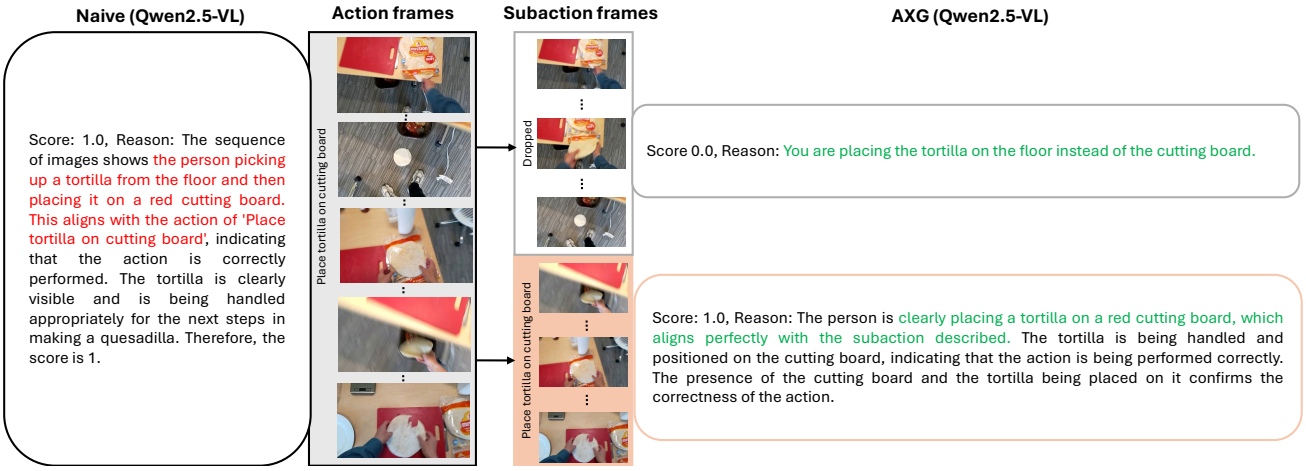

Figure S6. The qualitative result for action "place tortilla on cutting board" in task Quesadilla of EgoPER. The error is "the person drops tortilla on the floor at the middle of the action, discard it into trash, and place a new one."

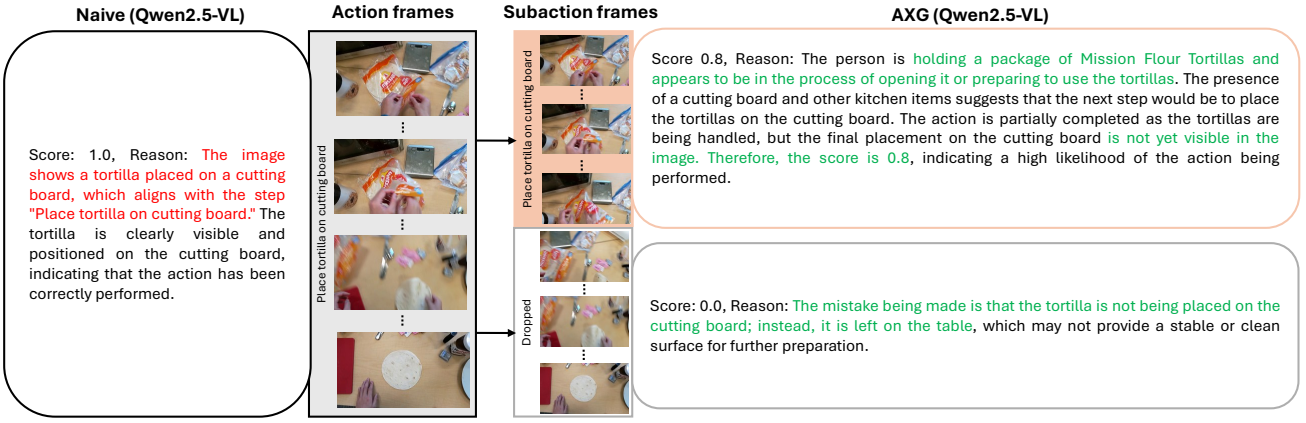

Figure S7. The qualitative result for action "place tortilla on cutting board" in task Quesadilla of EgoPER. The error is "the person places tortilla on the table instead of cutting board."

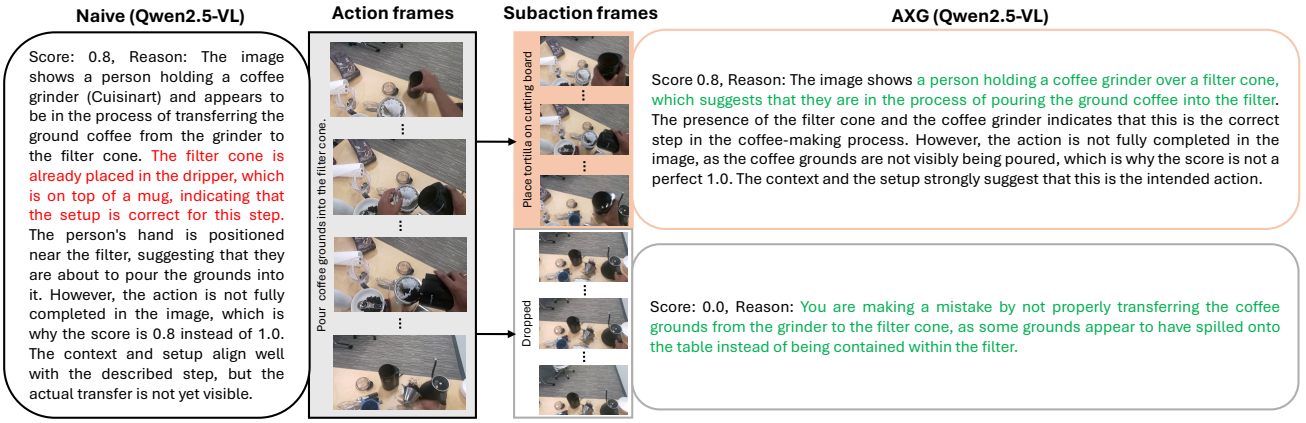

Figure S8. The qualitative result for action "pour coffee grounds into filter cone" in task Coffee of EgoPER. The error is "the person accidentally knock over the filter cone during the process."

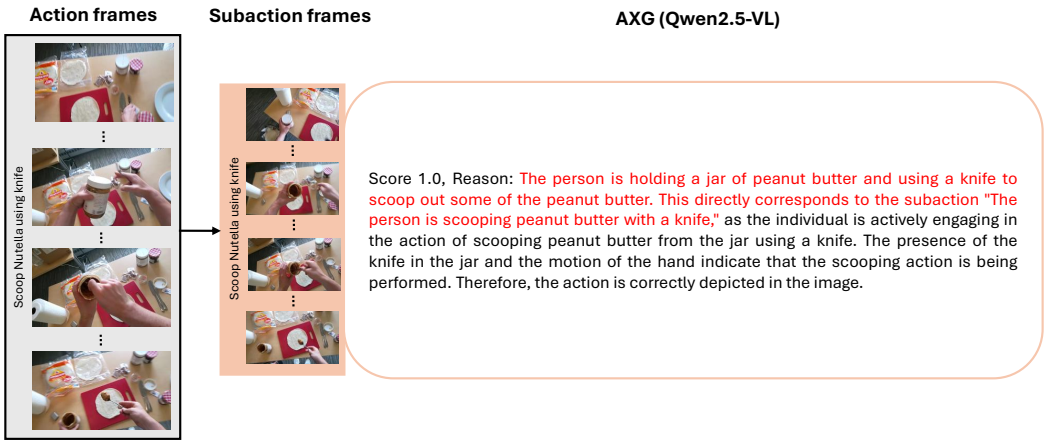

Figure S9. The qualitative result of a failure example for action "scoop peanut butter using knife" in task Pinwheel of EgoPER. The error is "scoop peanut butter using spoon instead of knife".

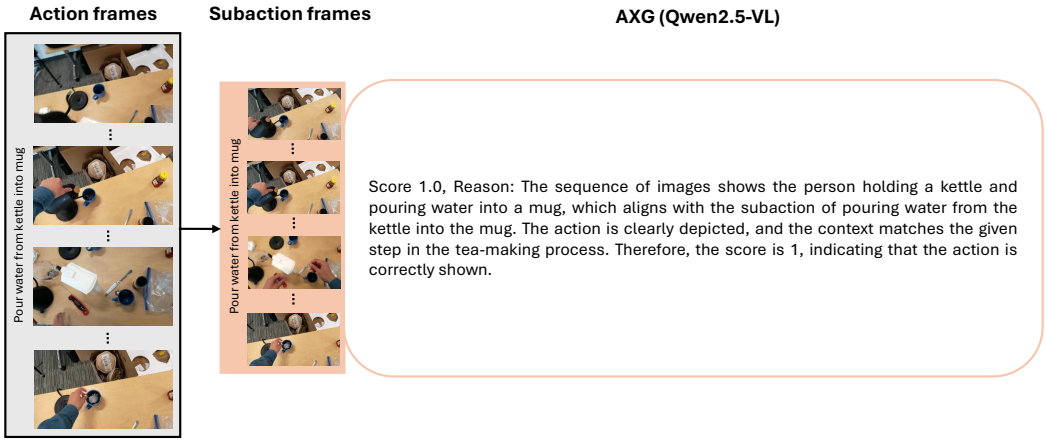

Figure S10. The qualitative result of a failure example for action "pour water from kettle into mug" in task Tea of EgoPER. The error is "the person pours water into a wrong mug that does not contain a tea bag which as been already placed in another mug".

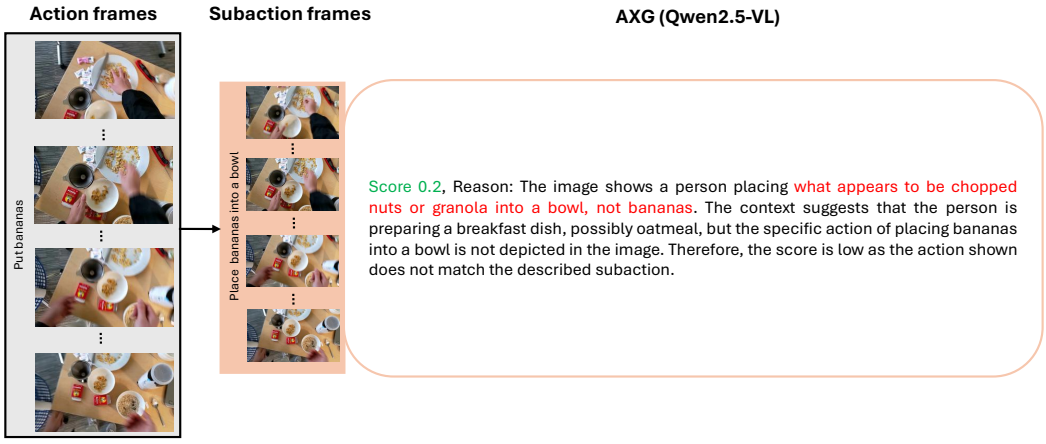

Figure S11. The qualitative result of a failure example for action "put bananas into bowl" in task Oatmeal of EgoPER. The error is "the person but bananas into a wrong bowl that does not contain oatmeal which as been already made in another bowl".