Table 6. Quantitative comparison of hallucination mitigation methods on the Dream1k [30] benchmark. **Bold** and underline, respectively, indicate the best and second best results achieved by these methods over the LLaVA-Video [39].

Method	Animation	Live action	Short	Stock	Youtube	Overall
LLaVA-Video [39]	27.6	31.4	33.4	36.7	33.0	32.5
+SFT [39]	19.9	27.7	28.3	36.0	33.0	29.2
+HALVA [26]	26.2	**32.3**	**34.6**	38.2	31.8	32.6
+HACL [12]	23.9	28.1	30.0	37.5	33.2	30.7
+SANTA (Ours)	**24.7**	31.0	31.8	**41.2**	**33.4**	**32.7**

7. Additional Experiment Setups

7.1. Training Data

Here, we provide a more detailed explanation of the three captioning subtasks in MiraData-9k [13]. The first is overall captioning, which provides a high-level summary of the video's main content. The second is main object captioning, which focuses on describing the primary objects present in the video. The last is background captioning, which provides contextual information about the surrounding environment.

We describe the process of obtaining the ground-truth action and object token set using GPT-4o [10] as parser, following the prompting templates presented in Fig. 8. As mentioned in 3.2, we parse the ground-truth captions to extract structured relation triples. Specifically, we employ GPT-4o [10] as a parser to derive each triple, consisting of an action verb and two associated object nouns, which we refer to as "related object instances" corresponding to the action verb. Thus, in our setup, N is set to 2. After extracting these ground-truth token sets, we use WordNet [7] to expand them by incorporating both synonyms (e.g., kid" for child" and get up" for stand") and hypernyms (e.g., person" for child" and move" for stand"). Since each video is described using three enriched captions, the extracted tokens comprehensively represent the visual facts depicted in the video. As a result, this augmentation process allows self-augmented hallucinative captions to capture the hallucination tendencies of the generated MLLM while ensuring that no visual information outside the video's content is introduced.

Tracklet-Phase Contrastive Alignment. As described in 3.3, to obtain the object tracklets, we employ Grounded-SAM2 [24, 25] to generate segmentation masks for each object instance, based on object information available in ground-truth captions. Using these masks, we crop the corresponding object regions, allowing us to identify which visual tokens contain object-specific information. These identified visual tokens are then used for region-level object alignment and relation-guided action alignment. The entire process is conducted offline.

7.2. Additional Evaluation Setups

In this subsection, we describe how we use MiraData-9k [13], a video-caption benchmark, to evaluate object and action hallucinations. As mentioned in Sec.7.1, we parse the ground-truth captions to extract sets of ground-truth object and action tokens. Using these sets, we then apply the hallucination evaluation metrics (i.e., Hal_{Act}, Hal_{Obj}, Cov_{Act}, Cov_{Obj}, F1_{Act}, and F1_{Obj} for both HalScore [4] and weighted-HalScore) introduced in Sec.4.1 to assess hallucination mitigation methods.

7.3. Comparisions

We compare *SANTA* to a simple baseline supervised fine-tuning baseline, and several the state-of-the-art visual hallucination mitigation methods, including DeCo [29], HALVA [26], and HACL [12]. For all methods, we utilize their publicly available codebases and adhere to their reported training configurations. To ensure a fair comparison, we align their training dataset, training and inference hyperparameter settings for all the methods, matching those used by LLaVA-Video [39] and Qwen2.5-VL [2].

8. Additional Quantitative Results

More Detailed Results on Video Description. Tab. 6 presents a quantitative comparison of various hallucination mitigation methods on Dream1k [30], a challenging video description benchmark comprising videos from five diverse sources: animation movies, live-action movies, TikTok-style videos, stock videos, and YouTube videos. The evaluation metric uses F1-score, calculated by the LLM-based evaluator, AutoDQ [30]. As shown in Tab. 6, *SANTA* achieves the highest overall F1-score, outperforming existing methods such as HALVA [26] and HACL [12]. This demonstrates that *SANTA* not only preserves but also enhances the video captioning capabilities of MLLMs. No-

Table 7. uantitative comparison of hallucination mitigation methods on the VideoMME [8] benchmark, with and without subtitles. **Bold** and **underline**, respectively, indicate the best and second best results achieved by these methods over the LLaVA-Video [39].

Method	Short		Medium		Long		Overall	
	with	without	with	without	with	without	with	without
LLaVA-Video [39]	80.1	77.0	66.6	61.8	58.3	52.1	68.3	63.6
+SFT [39]	54.8	56.8	45.4	45.9	35.3	35.0	45.2	45.9
+HACL [12]	73.1	69.9	62.3	57.9	53.2	51.4	62.9	59.7
+SANTA (Ours)	**76.8**	**72.3**	**64.6**	**59.8**	**55.4**	**49.6**	**65.6**	**60.6**

Table 8. Ablation of *SANTA* w/ the quality of object tracklets. We report the averaged F1 score of MiraData-9k [13] (HalFscore [4] and Weighted-HalFscore) to evaluate the effectiveness in mitigating both object and action hallucinations.

Method	HalFscore [4]		weighted-HalFscore	
	Avg. $F1_{Obj}$↑	Avg. $F1_{Act}$↑	Avg. $F1_{Obj}$↑	Avg. $F1_{Act}$↑
HACL [12]	36.5	28.5	52.8	30.5
SANTA w/ G-SAM2 (t=0.15)	37.2	29.4	54.0	31.6
SANTA w/ G-SAM2 (t=0.25)	**37.9**	**30.0**	**54.8**	**32.7**

tably, *SANTA* surpasses the original LLaVA-Video [39] baseline, further validating its effectiveness.

Results on General Video Question Answering. In Tab. 7, we present a quantitative comparison of different hallucination mitigation methods on the VideoMME [8] benchmark, evaluated under both with and without subtitles settings. VideoMME is a multiple-choice QA benchmark that spans a diverse range of video lengths, from short clips (11 seconds) to long-form content (up to 1 hour). The evaluation metric is accuracy, where higher scores indicate better video understanding performance.

As presented in Tab. 7, when applied to the same LLaVA-Video backbone, *SANTA* achieves higher QA accuracy than HACL [12], improving the overall performance by +2.7 (with subtitles) and +0.9 (without subtitles). However, both methods still exhibit a slight decrease compared to the original backbone, which is a shared limitation of current hallucination mitigation approaches. We hypothesize that this stems from using only video captioning data (e.g., MiraData-9k [13]) during post-training, without leveraging any video QA data. Thus, Appropriately incorporating a portion of this QA data could preserve the backbone's QA capability and, when combined with the hallucination reduction achieved by *SANTA*, further enhance overall performance beyond the original backbone.

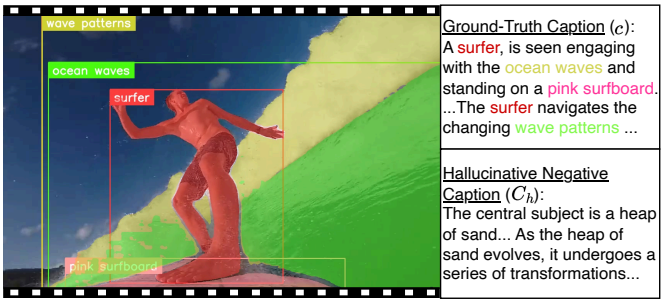

Figure 6. Qualitative results on hallucinative and object/action tracklets from MiraData-9k [13].

Analysis of Robustness with the Impact of Object Tracklet Quality. We use the SOTA video tracker Grounded-SAM2 [24, 25] to extract object tracklets. To analyze the robustness against possible tracking errors, we now ablate *SANTA* by training with noisy object tracklets, by lowering the detection confidence threshold (from default 0.25 to 0.15) to induce false positives on MiraData-9k [13].

As shown in Tab. 8, under the noisier setting (t=0.15) with more false positives, *SANTA* surpasses the existing SOTA hallucination mitigation method HACL [12] by a clear margin. Specifically, it achieves +2.1 and +0.6 absolute improvements on Avg. $F1_{Obj}$ and Avg. $F1_{Act}$ under HalFscore, and +2.9 and +1.1 under weighted-HalFscore, respectively. These results demonstrate that SANTA remains robust and effectively mitigates both object and action hallucinations, even when trained with imperfect object tracking results.

9. Additional Qualitative Results

Results on Hallucinative Negative and Object/Action Tracklets. Fig. 6 shows a surfing scene video frame, ground-truth caption c, and the hallucinative negative caption (C_h). In the visualized video frame, the presence of sea wave elements implies hallucination of "heap of sand...", etc. in C_h. The object tracklets are denoted as colored patches, and the action tracklets are derived from the associated two object tracklets (e.g., "stand" is associated with

"surfer" and "pink surfboard"). These relationships are determined by GPT4o [10] via prompting with the ground-truth caption, as described in Fig. 8.

Additional Results on MiraData-9k. In Fig. 7, we present additional qualitative results of applying *SANTA* with LLaAV-Video [39] on MiraData-9k [13], which exhibits object and action faithfulness within its generated captions.

Figure 7. Qualitative comparison of video captions predicted by HACL [12] and *SANTA* on MiraData-9k [13]. Note that words highlighted in green indicate action faithfulness, while those in red indicate action hallucination. Similarly, words in blue represent object faithfulness, whereas those in orange denote object hallucination.

Note that the string in each tuple must be a single word or a phrase that is exactly present in the original caption.
Follow the steps below carefully:
(1) Identify all objects mentioned in the caption that are both physically visible and clearly recognizable in the video.
[IMPORTANT]: Exclude any objects that are too abstract or cannot be visually recognized in the video (e.g., concepts like 'freedom' or 'thoughts')
(2) Identify all action verbs in the caption that are performed by the objects recognized in step (1).
[IMPORTANT]: Exclude any verbs that describe abstract actions, mental states, emotions, or general characteristics (e.g., 'highlight', 'believe', 'know', 'exist', 'indicate', 'attempt', 'try', 'engage' 'face' or 'suggest' must be excluded) as well as be verbs (e.g., 'is', 'are', 'was', 'were' must be excluded).
Only include verbs describing actions that can be observed visually in the video.
(3) For each action verb identified in step (2), find the corresponding object(s) recognized in step (1) that are performing or involved in the action.
Then, produce a tuple in the format (VERB, OBJECT1, OBJECT2), indicating that the VERB establishes a relationship between OBJECT1 and OBJECT2.
Note that every action verb and object in the tuples must be present in the list of objects and action verbs recognized in step (1) and (2).
Finally, combine all tuples into a single paragraph, ensuring the list of tuples is presented as continuous text without line breaks, bullet points, or any additional formatting.
<EXAMPLES>
Input:
A group of female basketball players practicing in a gym. The players are divided into two teams, wearing yellow and purple jerseys.
They are actively engaged in passing the ball, dribbling, and attempting shots.
The gym is well-lit with natural light streaming through large windows, and the wooden floor is marked with basketball court lines.
The players exhibit teamwork and coordination as they move dynamically across the court.
Output:
(1) List of Object(s): ['basketball players', 'gym', 'jerseys', 'ball', 'shots', 'windows', 'floor', 'basketball court lines']
(2) List of Action Verb(s): ['practicing', 'wearing', 'passing', 'dribbling', 'marked']
(3) List of Dependency Parsing Tuple(s): [('practicing', 'basketball players'), ('wearing', 'basketball players', 'jerseys'), ('passing', 'basketball players', 'balls'), ('dribbling', 'basketball players', 'balls'), ('marked', 'floor', 'basketball court lines')]
Input:
A first-person perspective of a motorcycle ride through a bustling cityscape, likely on a rainy day given the wet streets and overcast sky.
The rider maneuvers through various urban settings, including commercial areas filled with shops and billboards, and quieter residential zones.
The city is vibrant, with neon signs and advertisements adding color and life to the grey, damp environment.
The journey gives a dynamic view of city life from the unique vantage point of a motorcyclist, highlighting both the broad avenues and the more intimate alleyways of the urban landscape.
Output:
(1) List of Object(s): ['motorcycle', 'cityscape', 'rider', 'streets', 'sky', 'shops', 'billboards', 'neon signs', 'advertisements', 'motorcyclist', 'avenues', 'alleyways']
(2) List of Action Verb(s): ['ride', 'maneuvers']
(3) List of Dependency Parsing Tuple(s): [('ride', 'motorcycle', 'cityscape'), ('maneuvers', 'rider', 'shops'), ('maneuvers', 'rider', 'billboards')]
Input:
A virtual motorcycle ride through a bustling urban environment in a video game. The rider, clad in a yellow jacket, navigates the motorcycle through various cityscapes, including back alleys, main streets, and highways. The city is rich in detail, with dynamic weather conditions and a diverse array of pedestrians and vehicles populating the streets.
The journey captures the essence of a high-speed chase or a time-sensitive mission, with therider skillfully maneuvering around obstacles and traffic.
Output:
(1) List of Object(s): ['motorcycle', 'rider', 'urban environment', 'jacket', 'alleys', 'streets', 'highways', 'pedestrians', 'vehicles', 'streets', 'obstacles', 'traffic']
(2) List of Action Verb(s): ['ride', 'clad', 'navigates', 'populating', 'maneuvering']
(3) List of Dependency Parsing Tuple(s): [('ride', 'motorcycle', 'urban environment'), ('clad', 'rider', 'jacket'), ('navigate', 'rider', 'motorcycle'), ('populating', 'vehicles', 'streets'), ('populating', 'pedestrians', 'streets'), ('maneuver', 'rider', 'obstacles), ('maneuver', 'rider', 'traffic')]
Input:
Features towering skyscrapers adorned with bright advertisements and corporate logos, suggesting a commercial district in a highly developed urban setting.
The architecture displays a mix of high-tech design with sleek, metallic surfaces and occasional green spaces that add a touch of nature to the metallic urban environment.
The consistent rainfall and reflective wet surfaces enhance the night-time setting, emphasizing the city's vibrant nightlife and continuous activity.
Output:
(1) List of Object(s): ['skyscrapers', 'advertisements', 'logos', 'pedestrians', 'vehicles', 'streets', 'obstacles', 'traffic']
(2) List of Action Verb(s): ['adorned']
(3) List of Dependency Parsing Tuple(s): [('adorned', 'skyscrapers', 'advertisements'), ('adorned', 'skyscrapers', 'logos')]
<END_OF_EXAMPLES>

Now, perform dependency parsing on the given detailed caption according to the steps from (1) to (3).
Input: {GT_CAPTION}

Figure 8. The prompting template used to prompt GPT-4o [10] with few-shot demonstrations to perform the parsing task.