

# VDC-Agent: When Video Detailed Captioners Evolve Themselves via Agentic Self-Reflection

Qiang Wang<sup>1</sup>, Xinyuan Gao<sup>2</sup>, Songlin Dong<sup>3</sup>, Jizhou Han<sup>1</sup>, Jiangyang Li<sup>1</sup>, Yuhang He<sup>1</sup>, Yihong Gong<sup>1,3</sup>

<sup>1</sup>Xi'an Jiaotong University, Xi'an, China <sup>2</sup>Kuaishou Technology, Beijing, China

<sup>3</sup>Shenzhen University of Advanced Technology, Shenzhen, China

qwang@stu.xjtu.edu.cn, gaoxinyuan@kuaishou.com

## Abstract

We present **VDC-Agent**, a self-evolving framework for Video Detailed Captioning that requires neither human annotations nor larger teacher models. The agent forms a closed loop of caption generation, principle-guided scoring (score and textual suggestions), and prompt refinement. When caption quality regresses, a self-reflection path leverages the previous chain-of-thought to amend the update. Running this process on unlabeled videos produces trajectories of (caption, score) pairs. We convert the trajectories into preference tuples and filter out samples with JSON parsing errors, resulting in VDC-Agent-19K, which contains 18,886 automatically constructed pairs. We then fine-tune the base MLLM on this dataset using an easy-to-hard curriculum direct preference optimization. Built on Qwen2.5-VL-7B-Instruct, our **VDC-Agent-7B** attains state-of-the-art performance on the VDC benchmark with **49.08%** average accuracy and **2.50** score, surpassing specialized video captioners and improving over the base model by **+5.13%** accuracy and **+0.27** score at similar inference cost.

## 1. Introduction

With the explosive growth of online videos, understanding complex visual and temporal information has become a core capability for modern multimodal AI systems. Within this landscape, **Video Detailed Captioning (VDC)** [3, 16, 17, 21, 24, 27, 36, 42, 51] serves as a fundamental task that aims to generate fine-grained and comprehensive descriptions capturing objects, actions, interactions, and scene transitions in videos. By enabling a more precise and detailed understanding of video content, VDC facilitates progress in a wide range of applications such as visual question answering [19, 50], text-to-video generation [7, 13, 37], video retrieval [10], and temporal localization [26, 47].

Recent progress in VDC largely stems from fine-tuning Multimodal Large Language Models (MLLMs) on video-

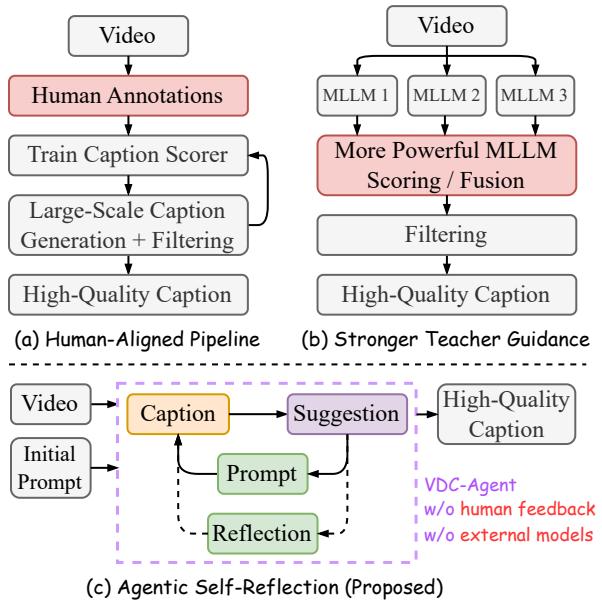


Figure 1. **Comparison of video captioning paradigms.** (a) Human-aligned pipelines rely on manual annotations to train caption scorers. (b) Multi-MLLM-based pipelines depend on multiple or stronger MLLMs for scoring or fusion. (c) Our proposed **VDC-Agent** achieves self-improvement through agentic self-reflection, requiring neither human annotations nor larger models.

caption datasets [29, 33, 38]. Although these methods have shown strong performance, they typically *rely on either distilling caption generation capabilities from more powerful MLLMs (proprietary models like GPT-4V [1] or open-source alternatives like Qwen-72B [2]) or incorporating extensive manual annotations for human preference alignment*, as shown in Fig. 1 (a)(b). For instance, ShareGPT4Video [4] constructs datasets using GPT-4V, followed by manual verification and filtering to enhance caption quality. Cockatiel [33] and OwlCap [49] rely on extensive human annotations to train caption scorers, or use more powerful captioning models to fuse captions from differ-

ent MLLMs. AVC-DPO [38] and VideoCap-R1 [29] build upon datasets constructed with powerful captioning models and employ reinforcement learning to further improve performance.

However, these approaches face several inherent limitations: prohibitive human annotation costs, access barriers to proprietary APIs, and substantial computational resources required for large-scale model inference. Consequently, enabling models to achieve autonomous reflection and iterative improvement in caption generation, without dependence on stronger MLLMs or extensive human annotations, has become critical for advancing beyond the current paradigm. To address this challenge, *we propose to treat the captioner itself as an autonomous agent that can generate, evaluate, and refine its own captions through iterative self-reflection.*

To this end, we propose *VDC-Agent*, a self-evolving video captioning framework that enables MLLMs to improve themselves through iterative self-reflection without requiring stronger external supervision. As depicted in Fig. 1 (c), VDC-Agent forms a closed-loop system that continuously refines its captioning ability by alternating between caption generation, evaluation, and prompt refinement. Given a collection of unlabeled videos, the model first generates captions using an initial prompt. It then conducts self-assessment based on a set of principles describing what constitutes a good caption (such as coverage of objects, actions, and temporal dynamics), assigns a quality score to the caption, and produces textual suggestions for improvement. These suggestions guide an internal *prompt refiner*, which updates the prompt in the next iteration. If the newly generated caption is even worse than the previous one, the model triggers a *self-reflection* mechanism that revisits the chain of thought used in the last prompt refinement, diagnosing why the previous update failed and avoiding the same mistake in subsequent steps. Through repeated cycles of caption–evaluation–refinement, our VDC-Agent can generate higher-quality video descriptions.

**To internalize VDC-Agent’s self-reflection capability into the MLLM**, enabling it to achieve the effect of multiple rounds of reflection within a single inference, we use VDC-Agent to collect a training set and fine-tune Qwen2.5-VL. In particular, we apply VDC-Agent to a high-resolution video corpus (Cockatiel-4K [33]), generating scored caption trajectories for each video and selecting the best and worst candidates to form a preference dataset, named VDC-Agent-19K, which comprises 18,886 preference pairs with corresponding score differences. To better exploit the high-quality captions produced by VDC-Agent, we introduce a curriculum Direct Preference Optimization (DPO) [31, 34] method for fine-tuning. Specifically, the score differences associated with each pair quantify how much caption quality varies across iterations. Unlike conventional DPO,

which treats all positive-negative pairs equally, we first prioritize pairs with larger score gaps to accelerate convergence, and then gradually incorporate smaller-gap pairs to learn more subtle distinctions between captions. We fine-tune *Qwen2.5-VL-7B-Instruct* with this curriculum strategy to obtain *VDC-Agent-7B*. Comprehensive experiments on the VDC benchmark across five dimensions show that our model achieves an average accuracy of **49.08%** and an average score of **2.50**, establishing new state-of-the-art performance. Compared to the base model, our approach yields average gains of **+5.13%** in accuracy and **+0.27** in score, validating the effectiveness of self-reflective evolution for video detailed captioning.

Our contributions can be summarized as follows:

- We propose VDC-Agent, an agentic self-evolving framework that lets a single MLLM generate, score, and refine video captions via principle-guided self-reflection, without human annotations or larger teacher models.
- We construct an preference dataset VDC-Agent-19K and introduce curriculum DPO that exploits the score gap between preferred and dispreferred captions as a difficulty signal, sampling from large to small gaps to enable easy-to-hard preference alignment for VDC.
- Built on Qwen2.5-VL-7B-Instruct, our VDC-Agent-7B achieves new state-of-the-art results on the VDC benchmark (49.08% accuracy, 2.50 score), surpassing prior video caption MLLMs and substantially improving over the base model at similar inference cost.

## 2. Related Work

### 2.1. Video Detailed Captioning

Video detailed captioning (VDC) [3, 21, 24, 32, 41] is a fundamental task in video understanding that aims to generate precise and comprehensive descriptions of video content. Early non-LLM approaches [11, 18] typically produced short, fragmentary captions with limited temporal grounding, which made them hard to use in downstream reasoning or retrieval systems. With the rapid progress of Multimodal Large Language Models (MLLMs), recent works have substantially improved the fluency and coherence of video captions. However, generic MLLMs are not explicitly optimized for fine-grained and temporally grounded descriptions, and they often miss subtle events, camera operations, or background cues that are critical for VDC.

Recent methods construct high-quality video-caption pairs and then fine-tune MLLMs on these curated datasets. Cockatiel [33] and Vriptor [43] build dense, human-annotated resources or human-preferred scorers to obtain detailed captions, while ShareGPT4Video [4], Shot2Story [12], and LLaVA-Video [48] leverage GPT-4V or GPT-4o to synthesize large-scale annotations that are further refined by filtering or human checks. More recently,

AVC-DPO [38] and VideoCap-R1 [29] employ powerful open-source models (*e.g.*, Qwen-72B) as teachers or reward providers to train smaller captioners with preference optimization or reinforcement learning. Despite their success, these approaches still hinge on stronger MLLMs or extensive human supervision for data construction, which leads to considerable annotation and computation costs and limits scalability and reproducibility in practice.

## 2.2. Multimodal Large Language Model Agent

Recent work has increasingly treated large language models as agents that interleave chain-of-thought reasoning with actions in an external environment, rather than as passive sequence-to-sequence predictors. Paradigms such as ReAct [45] and subsequent reflective agents [15, 35] encourage models to “think then act”, maintain textual memories, and update plans over multiple trials, enabling more robust decision making and error correction without necessarily updating model parameters. Extending this idea to the multimodal setting, MLLM agents take visual inputs and may call external tools while the model produces actions. MM-REACT [44] combines a conversational LLM with a pool of vision experts to solve complex visual understanding tasks through tool-augmented reasoning and action. CogAgent [14] and related GUI agents specialize in operating desktop or web interfaces from screenshots, while AppAgent [46] focuses on controlling smartphone applications by predicting tap and swipe actions from visual observations. VideoAgent [8, 40] employs an LLM-centric agent to query long videos and aggregate evidence for long-form video QA. Our VDC-Agent follows this agentic perspective but shifts the action from external environments to the model’s own prompts and intermediate captions, forming a closed loop of generation, principle-guided evaluation, and self-reflection that autonomously constructs preference data and strengthens a video detailed captioner without relying on larger teacher models.

## 3. Method

To obtain a stronger video detailed captioner, we first construct high-quality / low-quality caption pairs with an agentic data generator, **VDC-Agent**, and then fine-tune a multimodal large language model (MLLM) using direct preference optimization (DPO). Sec. 3.1 details how VDC-Agent iteratively improves caption quality. Sec. 3.2 shows how the resulting sequence of captions is transformed into a training dataset. Finally, Sec. 3.3 explains how the positive-negative pairs are leveraged to train our model via curriculum DPO.

### 3.1. VDC-Agent

Fig. 2 illustrates the overall framework of our VDC-Agent. Given a collection of videos, VDC-Agent automatically generates multiple captions for each video, scores their

quality, and finally produces preference pairs used for DPO training. The key idea is to let an MLLM iteratively refine the task prompts via self-reflection, so that later iterations yield more detailed captions.

We set a maximum number of iterations  $T$  and use  $t \in \{0, 1, \dots, T\}$  to index the interaction steps. At step  $t$ , the prompt, caption, score, suggestion are denoted by  $p_t$ ,  $y_t$ ,  $s_t$ , and  $g_t$ , respectively. As preparation, the user only needs to specify an initial prompt  $p_0$  (*e.g.*, “Please provide a detailed description of this video.”) and a set of textual principles  $R$  that describe what a good caption should look like (*e.g.*, covering camera motion, background, main objects, *etc.*). After that, VDC-Agent can run fully automatically over a video collection.

**Caption generation.** For a video  $x$  and a given prompt  $p_t$ , a base MLLM produces a candidate caption:

$$y_t = f(x; \Theta, p_t), \quad 0 \leq t \leq T, \quad (1)$$

where  $f(\cdot; \cdot)$  denotes the base MLLM and  $\Theta$  is its parameter set. When  $t = 0$ , the caption  $y_0$  is generated using the initial prompt  $p_0$ .

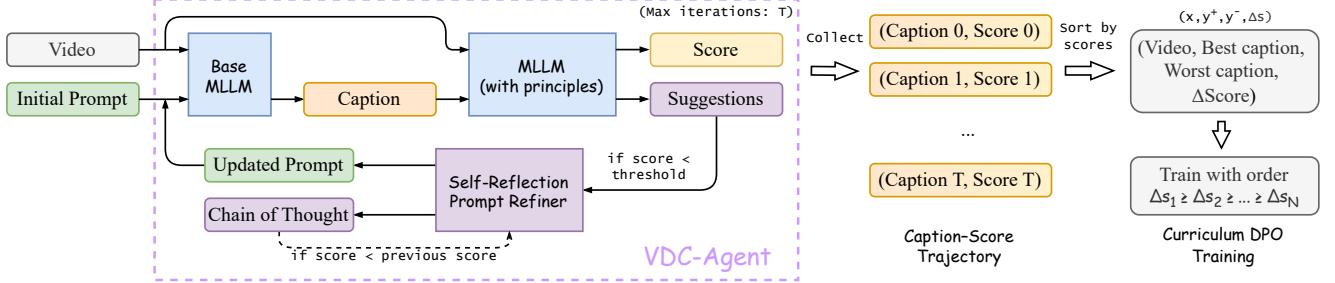
**Scoring and suggestion generation.** The caption  $y_t$  is then fed into the MLLM (which share the same backbone but is used with different instructions) together with the video  $x$  and the principle  $R$ . This *principle-guided* MLLM outputs both a scalar quality score and natural-language suggestions for improving the prompt:

$$(s_t, g_t) = f(x, y_t; \Theta, R), \quad 0 \leq t \leq T, \quad (2)$$

where  $s_t$  is the quality score from 0 to 100, and  $g_t$  is a textual suggestion describing how the next prompt should be revised (*e.g.*, asking for more spatial details or emphasizing the main object).

**Iterative prompt refinement with self-reflection.** VDC-Agent decides whether and how to update the prompt  $p_t$  based on the score  $s_t$ . Given a score threshold  $\lambda$ , if the caption already satisfies the requirement, *i.e.*,  $s_t \geq \lambda$ , the agent stops and outputs the current caption and score. Otherwise, it refines the prompt and continues.

We distinguish two cases for unsuccessful captions. If the score increases compared with the previous step, the agent simply performs *prompt refinement*, using the suggestion  $g_t$  as guidance. If the score even drops below the previous score, this indicates that the last refinement was harmful. In this case VDC-Agent triggers a *self-reflection* stage: it asks the MLLM to reason about why the previous refinement failed, using the current prompt and caption, together with the previous round’s prompt–caption pair and the chain-of-thought (CoT) produced by the previous prompt refiner, and then proposes a more reliable prompt update. Formally, the prompt update rule can be



**Figure 2. Overview of VDC-Agent and Dataset Construction.** Given a video  $x$  and an initial prompt  $p_0$ , the base MLLM produces a caption  $y_t$ . A principle-guided MLLM then returns a quality score  $s_t$  and textual suggestions  $g_t$ . If  $s_t \geq \lambda$ , the loop stops; otherwise the prompt is revised by the *Prompt Refiner*, and when  $s_t < s_{t-1}$  a *Self-Reflection* path is triggered using the previous chain-of-thought to diagnose and amend the update. This process repeats up to  $T$  iterations, yielding a set of (caption, score) pairs. We sort these pairs to obtain the best and worst captions ( $y^+$ ,  $y^-$ ) and compute the preference strength  $\Delta\text{Score} = s^+ - s^-$ . Each video thus forms a training tuple  $(x, y^+, y^-, \Delta s)$ , which is used to fine-tune the MLLM with *Curriculum DPO* via an easy-to-hard sampler (large to small  $\Delta s$ ).

written as:

$$p_{t+1} = \begin{cases} p_t, & s_t \geq \lambda, \\ f(y_t, s_t; \Theta, p_{\text{refine}}), & s_t \in [s_{t-1}, \lambda), \\ f(y_t, s_t, p_t; \Theta, p_{\text{reflect}}), & s_t < s_{t-1}, \end{cases} \quad (3)$$

where  $p_{\text{refine}}$  is the system-level instruction used by the prompt refiner and  $p_{\text{reflect}}$  is the instruction for the self-reflection module. For clarity, we omit the explicit CoT variables in Eq. (3); in practice, the CoT generated in the previous step is provided as additional context so that the agent can diagnose errors and avoid repeating them. During the above process, each iteration produces a pair  $(y_t, s_t)$ . After the loop terminates due to either reaching the threshold or hitting the maximum iteration  $T$ , we keep all caption-score pairs generated for this video and pass them to the data construction stage.

### 3.2. Dataset Construction by VDC-Agent

High-quality video data is crucial for training reliable detailed captioners. We adopt the Cockatiel-4K [33] corpus, which contains 4,008 high-resolution videos sampled from OpenVid-1M. To expose the model to complementary aspects of video understanding, we follow the taxonomy in VDC [3] and generate captions along five task dimensions: camera (shot type and camera motion), short (a concise summary), background (scene layout and context), main object (key objects and their attributes), and detailed (fine-grained temporal and spatial events). For each video  $x$  and each task dimension  $d$ , VDC-Agent executes the iterative procedure of Sec. 3.1 and produces a trajectory of caption-score pairs:

$$\mathcal{P}(x, d) = \{(y_t, s_t)\}_{t=0}^{T_v(x, d)}, \quad 1 \leq T_v(x, d) \leq T, \quad (4)$$

where  $y_t$  is the caption at iteration  $t$ ,  $s_t$  is its quality score, and  $T_v(x, d)$  is the number of iterations before termination

by threshold or by the cap  $T$ . Across 4,008 videos and 5 dimensions, we obtain a total of 20,040 raw caption sets.

**Rule-based filtering.** If a video already attains a score no smaller than the threshold  $\lambda$  in the first iteration (*i.e.*,  $|\mathcal{P}(x, d)| = 1$ ), the existing MLLM provides a sufficiently good caption without any refinement. Such cases offer little learning signal for preference modeling and are therefore removed. This filter discards 1,078  $(x, d)$  pairs. In addition, we remove 76 pairs due to JSON formatting / parsing errors during automatic generation. All filtering is *fully automatic* and rule-based, *requiring no human annotation*. After filtering, we retain 18,886 caption sets for the next stage.

**Constructing positive-negative pairs.** For every remaining set  $\mathcal{P}(x, d)$ , we sort captions by their scores and select the highest-scoring caption as the *best* caption  $y^+$  with score  $s^+$ , and the lowest-scoring caption as the *worst* caption  $y^-$  with score  $s^-$ . To quantify the preference strength (and later drive the curriculum), we compute the score gap  $\Delta s = s^+ - s^-$ . Each video and task dimension then contributes one training tuple  $(x, y^+, y^-, \Delta s)$ , forming the dataset used by the training in Sec. 3.3.

### 3.3. VDC Learning with Curriculum DPO

We fine-tune the base MLLM with DPO, which updates a policy  $\pi_\theta$  from paired preferences without training a reward model or running on-policy RL.

**Vanilla DPO.** Given a video  $x$ , a preferred caption  $y^+$  and a dispreferred caption  $y^-$ , DPO maximizes the probability that  $\pi_\theta$  prefers  $y^+$  over  $y^-$  while keeping it close to a reference policy  $\pi_{\text{ref}}$ :

$$\begin{aligned} \mathcal{L}_{\text{DPO}}(\theta; x, y^+, y^-) = \\ - \log \sigma \left( \beta \left[ \log \frac{\pi_\theta(y^+ | x)}{\pi_\theta(y^- | x)} - \log \frac{\pi_{\text{ref}}(y^+ | x)}{\pi_{\text{ref}}(y^- | x)} \right] \right), \end{aligned} \quad (5)$$

where  $\beta > 0$  controls the implicit KL strength.

**Why curriculum?** Vanilla DPO treats every pair equally, which is suboptimal for our data. In our setting, different

pairs exhibit different levels of difficulty: some preferred-dispreferred pairs differ markedly (large semantic gap), which are ideal for early-stage coarse adaptation; others are much closer (small gap), which are valuable for late-stage fine-grained alignment. Fortunately, the dataset produced by VDC-Agent naturally provides a difficulty signal, *i.e.*, the score gap  $\Delta_s$ , quantifying how much better the preferred caption is than the dispreferred one.

**Curriculum DPO.** To let the model learn *from easy to hard* in an optimization-friendly manner, we adopt a simple yet effective curriculum DPO. We exploit the score gap  $\Delta_s$  as a built-in difficulty signal and only modify the sampling order. Specifically, we sort all tuples  $\mathcal{D} = \{(x_i, y_i^+, y_i^-, \Delta_{s_i})\}_{i=1}^N$  by  $\Delta_{s_i}$  in descending order ( $\Delta_{s_1} \geq \dots \geq \Delta_{s_N}$ ), and feed mini-batches sequentially along this list (large  $\Delta_s \rightarrow$  small  $\Delta_s$ ). The optimization objective is

$$\min_{\theta} \sum_{i=1}^N \mathcal{L}_{\text{DPO}}(\theta; x_i, y_i^+, y_i^-), \text{ with order } i = 1 \rightarrow N. \quad (6)$$

Intuitively, large-gap pairs yield strong, low-variance gradients that quickly steer the policy toward the correct global behavior; small-gap pairs then refine subtle distinctions without delicate weight tuning. Moreover, a cosine-decay learning-rate schedule naturally complements this curriculum: early confident pairs are learned with larger step sizes, while later ambiguous pairs are absorbed with smaller steps. Empirically, we find that this simple curriculum accelerates convergence and yields better final alignment for video detailed captioning.

## 4. Experiments

### 4.1. Experimental Setup

**Training data.** We construct our training dataset based on Cockatiel-4K [33] due to its high visual quality, which contains 4,008 high-resolution videos sampled from OpenVid-1M [30]. We only use the videos from Cockatiel-4K without captions. Importantly, there is no overlap between the training videos and those used in the VDC evaluation benchmarks. Following our VDC-Agent pipeline, we generate raw caption trajectories on five task dimensions (camera, short, background, main object, detailed), producing 20,040 raw sets in total. Applying the rule-based filtering in Sec. 3.2 to remove trivial one-step cases and malformed JSON outputs, we obtain 18,886 preference pairs for training, denoted as VDC-Agent-19K.

**Implementation details.** We use Qwen2.5-VL-7B-Instruct [2] as our base MLLM. The maximum number of iterations  $T$  is set to 4.  $\lambda$  is set to 90. We fine-tune with curriculum DPO using VDC-Agent-19K. For efficiency, only the LLM backbone is adapted with LoRA while all other parameters are frozen. We set the LoRA rank and alpha to

16 and 32, respectively, with dropout 0.1. Training runs for 3 epochs with a cosine-decay learning-rate schedule (initial LR  $5 \times 10^{-5}$ , warmup 10% of total steps). We train on 4× NVIDIA A800 GPUs with a global batch size of 16.

**Evaluation benchmarks.** We evaluate on VDC, a video detailed captioning benchmark containing 1,027 videos. Following AuroraCap [3], we report the VDCscore as the principal metric to assess the quality of captions.

## 4.2. Main Results

**Compared methods.** We compare against a set of models with roughly 7B parameters (plus one commercial system) and organize them into two groups in Tab. 1. *General MLLMs* (top block) are versatile vision–language models, including *Llama 3.1-8B*, *Gemini 1.5 Pro*, *LLaMA-VID-7B*, *Video-ChatGPT-7B*, *Video-LLaVA-7B*, *LLaVA-OneVision-7B*, *VideoChat-Flash-7B*, and *Video-R1-7B*; since these models are not specifically trained for video captioning, their results are provided mainly for reference. *Video Caption MLLMs* (middle block) are designed for video captioning, such as *ShareGPT4Video-8B*, *Vriptor*, *AuroraCap-7B*, *Cockatiel-8B*, *VideoCap-R1-7B*, *SynPO*, *AVC-DPO-7B*, and *OwlCap-7B*, which are our primary points of comparison. Among them, *AVC-DPO* and *VideoCap-R1* leverage *Qwen2.5-72B* as an auxiliary model, potentially benefiting from larger capacity and prior knowledge; hence we mainly compare our method with the baseline *Qwen2.5-VL-7B-Instruct* (last block).

**Overall performance.** As summarized in Tab. 1, VDC-Agent-7B attains the best average VDCscore among all methods, with an average accuracy of 49.08 and an average score of 2.50. Even though it relies solely on self-reflective iterative optimization built on the baseline model, our method surpasses strong caption-focused systems, including *OwlCap-7B* (average 46.90/2.40) and *AVC-DPO-7B* (average 47.70/2.47). Most importantly, under the same backbone and training budget, our approach significantly improves over the base model *Qwen2.5-VL-7B-Instruct* by **+5.13** accuracy and **+0.27** score on average, validating the effectiveness of agentic self-refinement.

**Dimension-wise analysis.** Relative to Qwen2.5-VL-7B-Instruct, our VDC-Agent-7B yields consistent gains across all five dimensions: camera (+7.91 Acc / +0.49 Score), short (+1.73 / +0.08), background (+7.83 / +0.40), main object (+4.23 / +0.18), and detailed (+3.96 / +0.20). We obtain the highest scores on four detail-oriented dimensions (camera, background, main object, and detailed), which indicates stronger video understanding ability of spatial layout, salient entities, and fine-grained temporal events. While our model is not top-ranked on the short dimension, it emphasizes concise clip-level summarization that many general MLLMs already handle well; in contrast, our method pri-

Table 1. **VDCscore comparison on the VDC benchmark.** “Human” indicates whether human annotations are used. “External” indicates whether larger external teacher models are involved. We report Accuracy/Score for five dimensions (higher is better) and the average across dimensions. Qwen2.5-VL-7B-Instruct is the baseline model of our VDC-Agent-7B. **Bold** denotes the best results, and underline denotes the second-best results.

Model	Human	External	Camera Acc/Score	Short Acc/Score	Background Acc/Score	Main Object Acc/Score	Detailed Acc/Score	Average Acc/Score
<b>General MLLMs</b>								
Llama 3.1-8B [6]	-	-	17.83/1.00	17.90/1.02	19.52/1.00	19.57/1.10	20.10/1.22	18.98/1.07
Gemini 1.5 Pro [39]	-	-	38.68/2.05	35.71/1.85	43.84/2.23	47.32/2.41	43.11/2.22	41.73/2.15
LLaMA-VID-7B [23]	-	-	39.47/2.10	29.92/1.56	28.01/1.45	31.24/1.59	25.67/1.38	30.86/1.62
Video-ChatGPT-7B [28]	-	-	37.46/2.00	29.36/1.56	33.68/1.70	30.47/1.60	24.61/1.26	31.12/1.62
Video-LLaVA-7B [25]	-	-	37.48/1.97	30.67/1.63	32.50/1.70	36.01/1.85	27.36/1.43	32.80/1.72
LLaVA-OneVision-7B [20]	-	-	37.82/2.02	32.58/1.70	37.43/1.92	38.21/1.96	41.20/2.13	37.45/1.95
VideoChat-Flash-7B [22]	-	-	43.70/2.30	33.70/1.70	45.10/2.30	47.60/2.40	44.50/2.30	42.92/2.20
Video-R1-7B [9]	-	-	42.70/2.20	<b>44.50/2.30</b>	40.60/2.10	45.90/2.30	45.60/2.40	43.86/2.26
<b>Video Caption MLLMs</b>								
ShareGPT4Video-8B [4]	✗	✓	33.28/1.76	39.08/1.94	35.77/1.81	37.12/1.89	35.62/1.84	36.17/1.85
Vriptor [43]	✓	✗	37.64/1.96	38.35/2.00	37.11/1.94	37.02/1.93	38.49/2.00	37.72/1.97
AuroraCap-7B [3]	✗	✗	43.50/2.27	32.07/1.68	35.92/1.84	39.02/1.97	41.30/2.15	38.36/1.98
Cockatiel-8B [33]	✓	✗	42.25/2.19	<u>44.01/2.27</u>	43.89/2.26	43.85/2.26	44.00/2.27	43.60/2.25
VideoCap-R1-7B [29]	✗	✓	41.70/2.30	35.20/1.90	47.20/2.50	47.00/2.50	43.80/2.40	42.98/2.32
SynPO [5]	✗	✗	-/1.78	-/1.94	-/1.91	-/1.87	-/2.04	-/1.91
AVC-DPO-7B [38]	✗	✓	<u>50.40/2.66</u>	39.00/2.03	<u>49.90/2.57</u>	<u>50.50/2.58</u>	<u>48.90/2.54</u>	<b>47.70/2.47</b>
OwlCap-7B [49]	✗	✓	41.30/2.20	42.20/ <b>2.30</b>	41.40/2.10	45.20/2.30	43.40/2.30	46.90/2.40
Qwen2.5-VL-7B-Instruct [2]	-	-	42.61/2.18	37.76/1.91	44.10/2.22	49.00/2.47	46.25/2.35	43.95/2.23
VDC-Agent-7B (Ours)	✗	✗	<b>50.52/2.67</b>	39.49/1.99	<b>51.93/2.62</b>	<b>53.23/2.65</b>	<b>50.21/2.55</b>	<b>49.08/2.50</b>
<i>Improvement over Qwen2.5</i>			(+7.91/+0.49)	(+1.73/+0.08)	(+7.83/+0.40)	(+4.23/+0.18)	(+3.96/+0.20)	(+5.13/+0.27)

orizes improving richer, detail-centric dimensions that ultimately drive the best overall VDCscore.

### 4.3. Ablation and Discussion

**Why self-reflection effective?** To better understand why our agentic self-reflection design is necessary and how each component contributes, we conduct ablation experiments on the VDC benchmark. We compare four variants:

- **Baseline:** the original Qwen2.5-VL-7B-Instruct model without any additional mechanism;
- **Baseline+Principle:** a naive variant that directly concatenates the textual principles into the prompt without agentic iteration;
- **Baseline+Principle+Reflection:** which performs self-reflective refinement directly on the test set, but this leads to longer inference time.
- **VDC-Agent:** the proposed method that is trained on the dataset generated through self-reflection and then evaluated on the VDC benchmark.

As shown in Tab. 2, simply injecting the principle text (Baseline+P) improves over the baseline, demonstrating that explicit quality criteria provide useful prior knowledge. However, this naive strategy not only increases inference time but also fails to adapt to diverse video inputs due to its

fixed principle, resulting in only a modest +2.52% average accuracy gain. The third variant (Baseline+P+R) further enhances performance, validating that iterative diagnosis and correction indeed refine caption quality. Nevertheless, it assumes test-time adaptation and significantly prolongs inference time, which is impractical for deployment. In contrast, our VDC-Agent achieves comparable or better performance without accessing the test data, as the self-reflection mechanism is leveraged during training through agentic data generation. This demonstrates that VDC-Agent successfully distills the benefits of reflective refinement into a single forward model, providing both robustness and efficiency at inference time.

**Ablation on curriculum DPO.** To verify the contribution of the curriculum DPO in our method, we conduct an ablation by comparing three variants of our model under identical settings, as shown in Tab. 3: (1) VDC-Agent with Supervised Fine-Tuning (SFT), which is fine-tuned via standard supervised learning using the positive captions only; (2) VDC-Agent with Vanilla Direct Preference Optimization (DPO), which adopts Eq. (5) without curriculum scheduling; and (3) VDC-Agent with curriculum DPO, our full method with the proposed easy-to-hard sampling strategy guided by the score gap  $\Delta_s$ .

**Table 2. Ablation on self-reflection and principle components.** Baseline is Qwen2.5-VL-7B-Instruct, **P** denotes Principle, and **R** denotes self-Reflection. We compare the baseline, naive principle-augmented input, test-time self-reflection, and our VDC-Agent. While adding principles helps, it increases inference cost and fails to generalize to unseen videos. Self-reflection further improves caption quality but is impractical at inference time. Our model achieves the best balance of accuracy and efficiency by internalizing self-reflection through training-time agentic data generation.

Method	Camera		Short		Background		Main Object		Detailed		Average Acc Score	Inference Time	
	Acc	Score											
Baseline	42.61	2.18	37.76	1.91	44.10	2.22	49.00	2.47	46.25	2.35	43.95	2.23	15.5s
Baseline + P	47.76	2.43	38.81	1.96	47.82	2.41	50.07	2.53	47.87	2.42	46.47	2.35	22.3s
Baseline + P + R	45.15	2.29	39.13	1.98	50.02	2.53	52.13	2.63	49.09	2.49	47.10	2.38	164.9s
VDC-Agent	<b>50.52</b>	<b>2.67</b>	<b>39.49</b>	<b>1.99</b>	<b>51.93</b>	<b>2.62</b>	<b>53.23</b>	<b>2.65</b>	<b>50.21</b>	<b>2.55</b>	<b>49.08</b>	<b>2.50</b>	15.5s

**Table 3. Ablation on curriculum DPO.** We compare fine-tuning strategies on the VDC benchmark. SFT denotes Supervised Fine-Tuning, and DPO denotes Direct Preference Optimization. Curriculum strategy is not applicable to SFT due to dependence on  $\Delta s$ .

Method	Camera		Short		Background		Main Object		Detailed		Average Acc Score	
	Acc	Score	Acc	Score	Acc	Score	Acc	Score	Acc	Score		
VDC-Agent (SFT)	45.43	2.36	39.30	1.99	51.35	2.56	51.60	2.58	50.01	2.54	47.54	2.41
VDC-Agent (DPO)	47.95	2.44	38.79	1.97	51.66	2.58	52.38	2.62	49.36	2.49	48.03	2.42
VDC-Agent (Curriculum DPO)	50.52	2.67	39.49	1.99	51.93	2.62	53.23	2.65	50.21	2.55	49.08	2.50

**Table 4. Ablation on the robustness to principles.** Three contributors independently write distinct principle sets (P1–P3). R denotes self-reflection (please see Tab. 2). We compared the average accuracy and score on the VDC benchmark. Although minor variations exist, the final results of VDC-Agent remain highly stable across all versions, indicating strong robustness to principle.

Contributor	Method	Acc	Score
-	Baseline	43.95	2.23
1	Baseline + P1	46.47	2.35
	Baseline + P1 + R	47.10	2.38
	VDC-Agent (with P1)	49.08	2.50
2	Baseline + P2	45.83	2.32
	Baseline + P2 + R	46.55	2.36
	VDC-Agent (with P2)	48.84	2.48
3	Baseline + P3	46.85	2.37
	Baseline + P3 + R	47.25	2.39
	VDC-Agent (with P3)	49.02	2.50

**Robustness to principles.** In our VDC-Agent, principles act as textual guidelines that steer automatic prompt optimization. A natural question is whether our performance depend on the exact wording or content of these principles. To verify the robustness, we conducted an ablation study to examine the *sensitivity* of VDC-Agent to different principle sets. We invited three independent contributors to write their own versions of the principles, denoted as P1, P2, and P3, without mutual discussion. Each contributor wrote five principles for the five task dimensions (camera, short, background, main object, and detailed) with differing linguistic

**Table 5. Ablation study of the maximum iteration  $T$ .** We report the average accuracy and score on VDC. *Generation Time* is the average / sum wall-clock time VDC-Agent spends for generating the dataset. The reported time refers to the runtime on a single A800 GPU, and we have implemented code to easily enable parallel processing. Increasing  $T$  improves performance but also lengthens training data generation time, reflecting a compute-quality trade-off. We set  $T = 4$  by default.

Max Iterations $T$	Acc	Score	Generation Time
2	46.98	2.38	41.6s / 231.6h
3	48.15	2.43	58.3s / 324.5h
4	49.08	2.50	70.7s / 393.6h
5	49.29	2.52	85.5s / 476.0h
6	49.55	2.53	103.8s / 577.8h

styles, granularities, and emphases. For each set, we re-ran the complete pipeline in Sec. 3.1–Sec. 3.3, generated preference pairs, and trained with curriculum DPO on the same video dataset. As shown in Tab. 4, despite minor fluctuations across contributors, the final results remain highly consistent (around 49.0 accuracy and 2.5 score). This indicates that our improvement mainly stems from the agent’s iterative self-reflection and prompt refinement mechanism rather than the particular phrasing of principles, demonstrating strong robustness to principle design.

**Effect of the maximum iteration  $T$ .** VDC-Agent’s refinement depth is controlled by the hyperparameter  $T$ , which caps the number of agent loops. As shown in Tab. 5, performance improves steadily from the baseline (no agent) to  $T=4$ , where accuracy and score largely sat-



#### *Qwen2.5-VL-7B-Instruct*

In a spacious greenhouse with large glass windows allowing ample natural light, two individuals dressed in blue lab coats are engaged in a meticulous plant measurement activity. The greenhouse is filled with rows of potted tomato plants, each carefully placed on black trays. One person, holding a yellow tape measure, measures the height of a young tomato plant while the other assists by holding it steady. The environment is organized, with blue plastic crates and metal shelving units holding additional supplies and tools, emphasizing a professional and systematic approach to plant care and research.

#### *Video-R1-7B*

Inside a bright greenhouse, a researcher in a light blue lab coat stands beside long rows of potted plants arranged neatly on tables. Holding a yellow measuring tape, the researcher bends down to check how tall some of the plants have grown and sometimes writes the numbers in a small notebook. The camera follows as the researcher moves slowly along the row, inspecting the leaves and occasionally adjusting the position of a pot. Sunlight comes through the glass walls, and the overall scene looks like a careful experiment to monitor plant growth in a controlled environment.

#### *Cockatiel-13B*

The video begins with a view of a greenhouse filled with rows of potted plants. The camera pans across the greenhouse, showing various types of plants in white pots. A person wearing a blue shirt is seen tending to the plants, using a yellow measuring tape to measure the height of the plants. The person carefully measures each plant, taking note of their growth. The greenhouse is well-lit with natural light, and the plants appear healthy and well-maintained.

#### *VDC-Agent-7B (Ours)*

In a spacious, well-lit greenhouse filled with rows of potted plants, two researchers, dressed in teal lab coats, are meticulously measuring and tagging the plants. The greenhouse is structured with a high ceiling and large glass panels that allow ample natural light, creating an ideal environment for plant growth. The researchers work collaboratively, with one holding a yellow measuring tape while the other assists by holding the plant steady. They focus intently on a particular plant, ensuring precise measurements. The plant is positioned carefully on a black tray, surrounded by similar trays holding numerous other plants, all varying in height and stage of growth. The background reveals additional trays and a blue storage bin. The researchers' focus remains unwavering as they navigate through the rows. The overall atmosphere is one of focused productivity, with the researchers fully immersed in their task.

Figure 3. Qualitative comparison on a video from the VDC benchmark. We compare captions generated by *Qwen2.5-VL-7B-Instruct*, *Video-R1-7B*, *Cockatiel-13B*, and our *VDC-Agent-7B* under the same prompt: “Please provide a detailed description of the given video.” Text in green denotes correct, video-grounded descriptions, while red indicates incorrect or hallucinated content. Blue highlights fine-grained details that are correctly identified only by our *VDC-Agent-7B*.

urate (49.08/2.50) while keeping generation time reasonable. Increasing  $T$  beyond 4 yields marginal gains (*e.g.*,  $T=6$ : 49.55/2.53) at the cost of additional latency, indicating a practical efficiency–performance trade-off. At the same time, this highlights our method’s compute flexibility: users with ample resources can obtain higher caption quality and more powerful captioner by increasing  $T$ , while  $T=4$  remains a well-balanced default.

#### 4.4. Qualitative Results

Fig. 3 presents a qualitative comparison between *VDC-Agent-7B* and several strong baselines on a greenhouse video. The base model *Qwen2.5-VL-7B-Instruct* correctly recognizes the greenhouse scene and the plant-measuring activity, but its caption exhibits hallucination by arbitrarily assuming the plants to be tomato plants despite no such evidence in the video. *Video-R1-7B* and *Cockatiel-13B* provide more elaborate descriptions, yet they still miss background details (*e.g.*, the arrangement of trays and equipment) and camera motion (*e.g.*, the smooth pan along the greenhouse aisle as the researchers walk and measure the plants).

In contrast, our *VDC-Agent-7B* produces a more structured and comprehensive narrative that better captures the scene layout, the roles and interactions of entities, and the overall intent of the activity. The caption remains fluent while accurately depicting subtle attributes such as spatial organization, background context, and camera motion. These qualitative observations align with our quantita-

tive results, showing that agentic self-reflection enables the model to generate richer, more faithful, and better-grounded video detailed captions than both the base MLLM and existing captioning models.

## 5. Conclusion

We presented *VDC-Agent*, an agentic self-reflection framework that upgrades an off-the-shelf MLLM into a stronger video detailed captioner without human labels or larger teacher models. It couples principle-guided scoring with iterative prompt refinement and triggers reflection when updates regress, using past chain-of-thought to diagnose and correct errors. On unlabeled videos, this process produces caption–score trajectories that are transformed into positive–negative preferences and used, with the score gap, to drive curriculum DPO fine-tuning. Empirically, *VDC-Agent-7B* sets a new state-of-the-art on the VDC benchmark (average 49.08/2.50), surpassing prior caption-focused models and improving over its baseline at similar inference cost.

**Future Work.** In the future, we will extend our approach to larger backbones (*e.g.*, Qwen-32B) and broader video understanding tasks (*e.g.*, video question answering) to further investigate the performance ceilings, scalability, and universality of the proposed agentic framework.

## References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1
- [2] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 5, 6
- [3] Wenhao Chai, Enxin Song, Yilun Du, Chenlin Meng, Vashisht Madhavan, Omer Bar-Tal, Jenq-Neng Hwang, Saining Xie, and Christopher D Manning. Auroracap: Efficient, performant video detailed captioning and a new benchmark. *arXiv preprint arXiv:2410.03051*, 2024. 1, 2, 4, 5, 6
- [4] Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and generation with better captions. *Advances in Neural Information Processing Systems*, 37:19472–19495, 2024. 1, 2, 6
- [5] Jisheng Dang, Yizhou Zhang, Hao Ye, Teng Wang, Siming Chen, Hucheng Zheng, Yulan Guo, Jianhuang Lai, and Bin Hu. Synpo: Synergizing descriptiveness and preference optimization for video detailed captioning. *arXiv preprint arXiv:2506.00835*, 2025. 6
- [6] 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 e-prints*, pages arXiv–2407, 2024. 6
- [7] Tiehan Fan, Kepan Nan, Rui Xie, Penghao Zhou, Zhenheng Yang, Chaoyou Fu, Xiang Li, Jian Yang, and Ying Tai. Instancecap: Improving text-to-video generation via instance-aware structured caption. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 28974–28983, 2025. 1
- [8] Yue Fan, Xiaojian Ma, Rujie Wu, Yuntao Du, Jiaqi Li, Zhi Gao, and Qing Li. Videoagent: A memory-augmented multimodal agent for video understanding. In *European Conference on Computer Vision*, pages 75–92. Springer, 2024. 3
- [9] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025. 6
- [10] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*, pages 214–229. Springer, 2020. 1
- [11] Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2712–2719, 2013. 2
- [12] Mingfei Han, Linjie Yang, Xiaojun Chang, and Heng Wang. Shot2story20k: A new benchmark for comprehensive understanding of multi-shot videos. *arXiv preprint arXiv:2312.10300*, 2023. 2
- [13] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 1
- [14] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, et al. Cogagent: A visual language model for gui agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14281–14290, 2024. 3
- [15] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. Understanding the planning of llm agents: A survey. *arXiv preprint arXiv:2402.02716*, 2024. 3
- [16] Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. Video recap: Recursive captioning of hour-long videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18198–18208, 2024. 1
- [17] Minkuk Kim, Hyeon Bae Kim, Jinyoung Moon, Jinwoo Choi, and Seong Tae Kim. Do you remember? dense video captioning with cross-modal memory retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13894–13904, 2024. 1
- [18] Niveda Krishnamoorthy, Girish Malkarnenkar, Raymond Mooney, Kate Saenko, and Sergio Guadarrama. Generating natural-language video descriptions using text-mined knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, pages 541–547, 2013. 2
- [19] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. Tvqa: Localized, compositional video question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 1369–1379, 2018. 1
- [20] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 6
- [21] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhui Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 1, 2
- [22] Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, et al. Videochat-flash: Hierarchical compression for long-context video modeling. *arXiv preprint arXiv:2501.00574*, 2024. 6
- [23] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024. 6
- [24] Long Lian, Yifan Ding, Yunhao Ge, Sifei Liu, Hanzi Mao, Boyi Li, Marco Pavone, Ming-Yu Liu, Trevor Darrell, Adam

- Yala, et al. Describe anything: Detailed localized image and video captioning. *arXiv preprint arXiv:2504.16072*, 2025. 1, 2
- [25] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5971–5984, 2024. 6
- [26] Yi Liu, Haowen Hou, Fei Ma, Shiguang Ni, and Fei Richard Yu. Mllm-ta: Leveraging multimodal large language models for precise temporal video grounding. *IEEE Signal Processing Letters*, 2024. 1
- [27] HuiLan Luo, Xia Cai, and Lik-Kwan Shark. Frame-by-frame multi-object tracking-guided video captioning. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 1
- [28] Muhammad Maaz, Hanooona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602, 2024. 6
- [29] Desen Meng, Rui Huang, Zhilin Dai, Xinhao Li, Yifan Xu, Jun Zhang, Zhenpeng Huang, Meng Zhang, Lingshu Zhang, Yi Liu, et al. Videocap-r1: Enhancing mllms for video captioning via structured thinking. *arXiv preprint arXiv:2506.01725*, 2025. 1, 2, 3, 6
- [30] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024. 5
- [31] Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. Enhancing alignment using curriculum learning & ranked preferences. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12891–12907, 2024. 2
- [32] Iqra Qasim, Alexander Horsch, and Dilip Prasad. Dense video captioning: A survey of techniques, datasets and evaluation protocols. *ACM Computing Surveys*, 57(6):1–36, 2025. 2
- [33] Luozheng Qin, Zhiyu Tan, Mengping Yang, Xiaomeng Yang, and Hao Li. Cockatiel: Ensembling synthetic and human preferenced training for detailed video caption. *arXiv preprint arXiv:2503.09279*, 2025. 1, 2, 4, 5, 6
- [34] 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*, 36:53728–53741, 2023. 2
- [35] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023. 3
- [36] Peipei Song, Dan Guo, Xun Yang, Shengeng Tang, and Meng Wang. Emotional video captioning with vision-based emotion interpretation network. *IEEE Transactions on Image Processing*, 33:1122–1135, 2024. 1
- [37] Rui Sun, Yumin Zhang, Tejal Shah, Jiahao Sun, Shuoying Zhang, Wenqi Li, Haoran Duan, Bo Wei, and Rajiv Ranjan. From sora what we can see: A survey of text-to-video generation. *arXiv preprint arXiv:2405.10674*, 2024. 1
- [38] Jiyang Tang, Hengyi Li, Yifan Du, and Wayne Xin Zhao. Avc-dpo: Aligned video captioning via direct preference optimization. *arXiv preprint arXiv:2507.01492*, 2025. 1, 2, 3, 6
- [39] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 6
- [40] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *European Conference on Computer Vision*, pages 58–76. Springer, 2024. 3
- [41] Hongchen Wei, Zhihong Tan, Yaosi Hu, Chang Wen Chen, and Zhenzhong Chen. Longcaptioning: Unlocking the power of long video caption generation in large multimodal models. *arXiv preprint arXiv:2502.15393*, 2025. 2
- [42] Zeyu Xi, Ge Shi, Haoying Sun, Bowen Zhang, Shuyi Li, and Lifang Wu. Eika: Explicit & implicit knowledge-augmented network for entity-aware sports video captioning. *Expert Systems with Applications*, 274:126906, 2025. 1
- [43] Dongjie Yang, Suyuan Huang, Chengqiang Lu, Xiaodong Han, Haoxin Zhang, Yan Gao, Yao Hu, and Hai Zhao. Vript: A video is worth thousands of words. *Advances in Neural Information Processing Systems*, 37:57240–57261, 2024. 2, 6
- [44] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023. 3
- [45] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022. 3
- [46] Chi Zhang, Zhao Yang, Jiaxuan Liu, Yanda Li, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. Appagent: Multimodal agents as smartphone users. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20, 2025. 3
- [47] Quan Zhang, Jinwei Fang, Rui Yuan, Xi Tang, Yuxin Qi, Ke Zhang, and Chun Yuan. Weakly supervised temporal action localization via dual-prior collaborative learning guided by multimodal large language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24139–24148, 2025. 1
- [48] 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
- [49] Chunlin Zhong, Qiuxia Hou, Zhangjun Zhou, Shuang Hao, Haonan Lu, Yanhao Zhang, He Tang, and Xiang Bai. Owlcap: Harmonizing motion-detail for video captioning

- via hmd-270k and caption set equivalence reward. *arXiv preprint arXiv:2508.18634*, 2025. 1, 6
- [50] Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 6439–6455, 2022. 1
- [51] Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. Streaming dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18243–18252, 2024. 1