arXiv:2412.00624v1 [cs.CV] 1 Dec 2024

VideoSAVi: Self-Aligned Video Language Models without Human Supervision

Yogesh Kulkarni Pooyan Fazli
Arizona State University
{ykulka10, pooyan}@asu.edu
https://videosavi.github.io/

Abstract

Recent advances in vision-language models (VLMs) have significantly enhanced video understanding tasks. Instruction tuning (i.e., fine-tuning models on datasets of instructions paired with desired outputs) has been key to improving model performance. However, creating diverse instruction-tuning datasets is challenging due to high annotation costs and the complexity of capturing temporal information in videos. Existing approaches often rely on large language models to generate instruction-output pairs, which can limit diversity and lead to responses that lack grounding in the video content. To address this, we propose VideoSAVi (Self-Aligned Video Language Model), a novel self-training pipeline that enables VLMs to generate their own training data without extensive manual annotation. The process involves three stages: (1) generating diverse video-specific questions, (2) producing multiple candidate answers, and (3) evaluating these responses for alignment with the video content. This self-generated data is then used for direct preference optimization (DPO), allowing the model to refine its own high-quality outputs and improve alignment with video content. Our experiments demonstrate that even smaller models (0.5B and 7B parameters) can effectively use this self-training approach, outperforming previous methods and achieving results comparable to those trained on proprietary preference data. VideoSAVi shows significant improvements across multiple benchmarks: up to 28% on multi-choice QA, 8% on zero-shot open-ended QA, and 12% on temporal reasoning benchmarks. These results demonstrate the effectiveness of our self-training approach in enhancing video understanding while reducing dependence on proprietary models.

1. Introduction

Vision-language models (VLMs) [23, 31, 42] have made significant strides by integrating visual perception with the reasoning capabilities of large language models (LLMs) [13, 38, 39]. These models excel in interpreting and generating contextually relevant responses through the combination of image encoders and language generation techniques. Building on this foundation, recent video-large language models (Video-LLMs) [22, 29, 69] incorporate temporal dimensions, enabling comprehensive video understanding by transforming video frames into tokens that LLMs can process [25, 35]. While Video-LLMs demonstrate impressive capabilities in tasks, such as video captioning and question answering, they typically require vast, high-quality annotated datasets, making them resource-intensive and limiting their scalability.

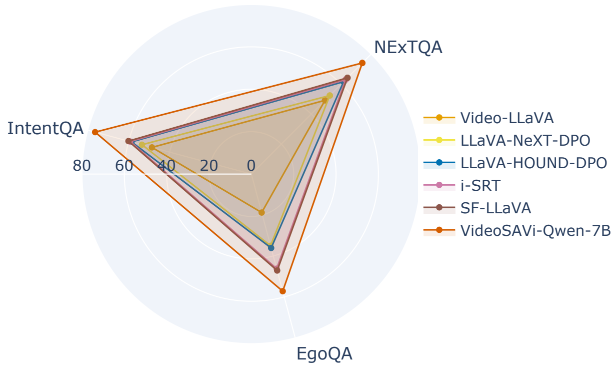


Figure 1. Comparative performance of selected methods on three multiple-choice QA benchmarks: (1) NExTQA, (2) EgoQA, and (3) IntentQA. VideoSAVi sets a new state of the art, demonstrating significant advancements in temporal reasoning and intent recognition for video understanding.

Instruction tuning, which involves training models to follow specific commands and generate appropriate responses, has been pivotal in advancing both VLMs and Video-LLMs [6, 31, 57, 62]. However, this approach faces significant challenges in the video domain due to the scarcity of large, high-quality training datasets. For instance, while image-based datasets encompass up to 500K instruction-response pairs [31], video instruction datasets such as VideoInstruct-100K [35] contain only around 13K unique videos. Creating extensive video instruction datasets, whether through manual annotation or LLM generation, incurs substantial costs. For instance, models such as GPT-

4V [37] can cost around $200 to generate just 6,000 image descriptions [12], with expenses rising further for more complex or longer video data. This dependence on extensive annotated data and usage of proprietary models often restricts the adaptability of Video-LLMs, posing a barrier to broader applications. Addressing this limitation, our study demonstrates that Video-LLMs can independently generate their own training data and align themselves with video content, thereby significantly reducing the dependency on large annotated datasets and enhancing scalability.

Recent research has focused on synthetic data generated by models, enabling the scalable, diverse, and low-cost creation of training data [1, 49]. This approach offers a promising alternative to manual annotations, allowing models to improve their understanding across new tasks via self-training. This leads us to two main research questions:

RQ1: How can we leverage synthetic data to enhance the performance of Video-LLMs in video understanding tasks without depending on expensive human annotations or proprietary APIs?

RQ2: How can we ensure that synthetic data aligns with video content to maintain the accuracy of model responses?

To address these questions, we propose VideoSAVi (Self-Aligned Video Language Model), a novel self-training pipeline that enables VLMs to generate and refine their own training data without extensive manual annotation or reliance on costly proprietary models. We ensure the generated synthetic data is aligned with video content through direct preference optimization (DPO) [43]. Current DPO methods rely on binary preferences [14], which do not fully capture the finer relationships between different responses. Further, initial iterations of self-training carry an inherent risk of favoring less accurate responses that do not align with the actual video content. To address these limitations, VideoSAVi introduces enhancements, such as CLIP filtering of responses, ensuring accurate and video-aligned responses.

Experiments demonstrate VideoSAVi's state-of-the-art (SOTA) ability to comprehend and reason over complex video content across diverse tasks such as multi-choice QA, open-ended QA, and temporal reasoning. Using a robust self-training pipeline with DPO for precise vision-language alignment, VideoSAVi achieves substantial benchmark performance improvements over baseline models. Additionally, iterative fine-tuning enables VideoSAVi to generate high-quality synthetic training data, extending its capabilities to novel prompts beyond its original training distribution. In summary, our contributions are as follows:

1. We present VideoSAVi, a novel self-training framework that generates synthetic data for fine-tuning Video-LLMs, significantly reducing the reliance on expensive human annotations or proprietary models.
2. We introduce CLIP-adjusted DPO, a novel optimization approach that extends traditional preference learning with visual similarity metrics to ensure video-grounded responses
3. We conduct extensive evaluations across multiple video understanding benchmarks, demonstrating significant improvements: 28% on multi-choice QA, 8% on zero-shot open-ended QA, and 12% on temporal reasoning benchmarks, outperforming models that rely on proprietary preference data for alignment through DPO.

Through this work, we address the dual challenges of enhancing the performance of Video-LLMs without incurring high annotation costs and ensuring the alignment of synthetic data with actual video content.

2. Related Work

2.1. Video-LLMs

Recent advancements in Video-LLMs have improved video understanding, with models such as Video-LLaVA [29], which uses unified visual representations, and Video-LLaMA [68], which integrates audio-visual modalities. However, these models are data- and computation-intensive, as they rely on large-scale datasets for feature alignment. Later models have focused on specific areas. LLaVA-NeXT-Video [71] improves zero-shot understanding without explicit video training but lacks fine-grained temporal reasoning. VTimeLLM [17] achieves moment-level temporal understanding through structured training but requires detailed annotations, limiting scalability. VideoLLM-Online [9] enables real-time dialogue with continuous video streams but demands high computational resources. Koala [52] uses key frame-conditioned processing for efficiency, though it may miss the continuous context needed for long videos. In contrast, our method, VideoSAVi, is a self-training framework that generates its own training data to improve vision-text alignment. By using DPO and iterative alignment with self-generated data, VideoSAVi reduces reliance on large datasets while achieving robust video understanding across various tasks.

2.2. Learning from AI Feedback

Reinforcement learning from human feedback (RLHF) is widely used to align LLMs with human preferences and ensure safer responses [7, 39, 50]. However, RLHF requires costly high-quality human-collected preference data. To address this, reinforcement learning from AI feedback (RLAIF) methods have been proposed [19, 51], which use AI-generated feedback instead of human feedback to guide model training. These methods typically use proximal policy optimization (PPO) [45] for training. For multimodal models, Ahn et al. [2] introduce the VLM-RLAIF framework to enhance video-text alignment. Unlike PPO-based approaches, we employ DPO [43], which directly optimizes

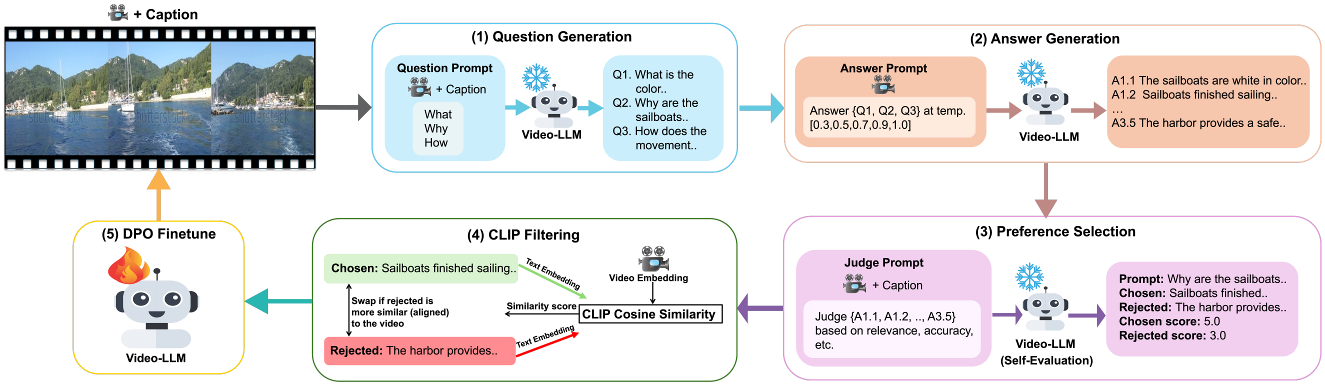

+ Caption
(1) Question Generation
Question Prompt
+ Caption
What
Why
How
Video-LLM
Q1. What is the color..
Q2. Why are the sailboats..
Q3. How does the movement..
(2) Answer Generation
Answer Prompt
Answer {Q1, Q2, Q3} at temp. [0.3,0.5,0.7,0.9,1.0]
Video-LLM
A1.1 The sailboats are white in color..
A1.2 Sailboats finished sailing..
...
A3.5 The harbor provides a safe..
(3) Preference Selection
Judge Prompt
+ Caption
Judge {A1.1, A1.2, .., A3.5} based on relevance, accuracy, etc.
Video-LLM (Self-Evaluation)
Prompt: Why are the sailboats..
Chosen: Sailboats finished..
Rejected: The harbor provides..
Chosen score: 5.0
Rejected score: 3.0
(4) CLIP Filtering
Chosen: Sailboats finished sailing..
Text Embedding
Video Embedding
Swap if rejected is more similar (aligned) to the video
Similarity score
CLIP Cosine Similarity
Rejected: The harbor provides..
Text Embedding
(5) DPO Finetune
Video-LLM

Figure 2. **Overview of VideoSAVi's self-training pipeline.** Starting with a baseline model trained on instruction data, our pipeline consists of five key components: (1) Question Generation: generates diverse "What", "Why", and "How" questions for each video using ground truth captions as context; (2) Answer Generation: produces five candidate answers at different temperatures [0.3, 0.5, 0.7, 0.9, 1.0] for each question; (3) Preference Selection: evaluates generated answers using a structured prompt template, resulting in scores from 1 to 5; (4) CLIP Filtering: refines the highest and lowest scored responses as positive/negative pairs using CLIP similarity scores to ensure video-text alignment; (5) DPO Finetune: optimizes the model using the filtered preference pairs. The model iteratively improves through a self-training process, where the DPO-tuned model from each iteration serves as the baseline for the subsequent iteration. 🔥 → trainable, ❄ → frozen.

preference learning without the need for reward modeling or policy optimization, offering a more streamlined alternative to traditional reinforcement learning methods. Recent work has applied DPO to Video-LLMs. For example, LLaVA-HOUND-DPO [70] and its extension i-SRT [1] use preference data generated from proprietary models such as GPT, with i-SRT adding iterative refinement for further performance gains. LLaVA-NeXT-DPO [30] follows a similar training recipe with a newer backbone (LLaVA-NeXT-Video [71]). In contrast, our work demonstrates that Video-LLMs can generate their own synthetic preference data and iteratively refine themselves for improved multimodal alignment.

2.3. Self-Training

Self-training has become a powerful method for improving language model performance [15, 18, 46, 53, 63, 66]. The main idea is for models to generate their own training data and use it to refine their performance iteratively, with notable success in enhancing reasoning and task-specific capabilities. Recent advances have applied self-training to vision-language models (VLMs) [12, 48, 49] and Video-LLMs [74]. However, these methods often rely purely on prompting [12] to generate preference data, use expensive GPT APIs [48], or depend on iterative learning from feedback with ground truth labels [74]. Additionally, directly prompting models to generate preference data can lead to performance issues due to the inherent instability of LLMs [64]. This creates a key research gap: developing synthetic preference data that can improve model performance without the limitations of current methods. Our work addresses this by combining self-questioning LLMs [49, 55] with model feedback mechanisms [2], enabling the self-training of a Video-LLM aligned with model-based preferences through DPO [43].

3. VideoSAVi

VideoSAVi is a Video-LLM that we trained using model-generated (synthetic) DPO preference data. Our improvements stem from two key aspects that address limitations in prior work. First, in vision-text alignment, previous approaches rely on GPT models' world knowledge [38] to generate DPO preference data [70] or use significantly more instruction-following data [2]. In contrast, we rely solely on our baseline model for prompt generation as well as for evaluating and selecting the accepted and rejected responses. Second, the chosen and rejected responses are not always grounded in actual video content. We address this by using CLIP [42] similarity scores and adapting the DPO loss function accordingly. As shown in Figure 2, our self-training pipeline enables VideoSAVi to iteratively improve through five key stages, from question generation to DPO finetuning.

3.1. Training Pipeline

Stage I: Supervised Fine-Tuning. We perform supervised fine-tuning (SFT) on the model using the instruction-following datasets such as LLaVA-Instruct [31] and VideoInstruct-100K [35]. This phase serves as a crucial warmup, enabling the Video-LLM to comprehend and respond to varied types of instructions. In our pipeline, this process trains the model to align effectively with the intended behavior for several of our core tasks: evaluating the response quality (i.e., acting as a judge), generating contextually relevant questions, and providing answers to these questions. After this stage, no additional instruction-following data is used.

Algorithm 1 Self-Training with CLIP-Adjusted DPO Loss

```
Input: Video datasets $D = \{D_{\text{VIDAL}}, D_{\text{Charades}}, D_{\text{WebVid}}\}$; For each $D' \in D$:
  • Videos $\{v_j^{D'}\}_{j=1}^{N_{D'}}$
  • Captions $\{c_j^{D'}\}_{j=1}^{N_{D'}}$
  Initial model $M$ and parameters $\theta^0$; Iterations $T$; Questions per video $K$; Learning rate $\eta$; Scaling factor $\beta$; Regularization $\lambda$
Output: Optimized model parameters $\theta^T$
for $t = 0$ to $T - 1$ do
    (1) Generate Questions
    for each $D' \in D$ do
        for $j = 1$ to $N_{D'}$ do
            for $k = 1$ to $K$ do
                Generate question $q_{j,k}$ using $M^t(c_j^{D'})$
            end for
        end for
    end for
    (2) Generate Answers
    for each $q_{j,k}$ do
        for $i = 1$ to 5 do
            Generate answer $a_{j,k,i}$ using $M^t(v_j^{D'}, q_{j,k})$
        end for
    end for
    (3) Evaluate and Select Preferences
    for each $q_{j,k}$ do
        for $i = 1$ to 5 do
            Compute score $s_i = M^t(v_j^{D'}, c_j^{D'}, q_{j,k}, a_{j,k,i})$
        end for
        $a_k^+ = a_{j,k,\arg\max_i s_i}, a_k^- = a_{j,k,\arg\min_i s_i}$
        (4) CLIP Filtering
        $c_k^+ = \text{CLIP}(v_j^{D'}, a_k^+), \quad c_k^- = \text{CLIP}(v_j^{D'}, a_k^-)$
        Assign $s_k = +1$ if $c_k^+ \geq c_k^-$ else $s_k = -1$
        Add $(v_j^{D'}, q_{j,k}, a_k^+, a_k^-, s_k)$ to $\mathcal{D}$
    end for
    (5) Finetune using DPO
    Adjusted DPO loss:
        $\mathcal{L}_{\text{Total}}(\theta) = -\sum_{\mathcal{D}} \Big[ \log \sigma \left( \beta s_k (\delta_\theta - \delta_{\text{ref}}) \right) + \lambda \log p_\theta(a_k^+ \mid v_j, q_k) \Big]$
    Update model parameters:
        $\theta^{t+1} \leftarrow \theta^t - \eta \nabla_\theta \mathcal{L}_{\text{Total}}(\theta^t)$
end for
return $\theta^T$
```

Stage II: Self-Training. Starting from the SFT model checkpoint obtained in Stage I, we refine the model's performance through iterative self-training on synthetic data. Utilizing three video datasets—VIDAL [73], STAR (Charades) [58], and WebVid [5]—we generate contextually relevant question-answer pairs. The model evaluates its own generated answers to produce preference data for DPO training. Through this iterative process, we progressively enhance the model's judgment quality and response accuracy. Detailed methodology is provided in the following section.

3.2. Self-Training

Following the initial SFT phase, we introduce an iterative refinement process to further optimize our model, as detailed in Algorithm 1. Starting with the model M^0 obtained from Stage I, we use it across all datasets $D \in \{D_{\text{VIDAL}}, D_{\text{Charades}}, D_{\text{WebVid}}\}$. For each video $v_j \in D$, we generate synthetic question-answer pairs (q_k, a_k) using the ground truth captions c_j as context. The model M^0 also serves as a judge to evaluate the quality of these generated answers, producing quality scores s_k. We select the answer with the highest score as the positive example a_k^+ and the answer with the lowest score as the negative example a_k^-.

To enhance the robustness of our preference data for DPO, we implement a CLIP-based adjustment mechanism. For each pair (a_k^+, a_k^-), we compute the CLIP similarity scores between the video v_j and each answer, and determine a per-example sign s_k to adjust the DPO loss accordingly. The model then undergoes iterative self-training, updating the model parameters θ^{t+1} by minimizing the adjusted DPO loss over the collected preferences from iteration t. The per-example sign s_k adjusts the loss to account for cases where the CLIP model indicates that the negative example is more aligned with the video content than the positive example, ensuring that the DPO training process favors the response that is better aligned with the video content. This approach aligns video and text modalities within a unified framework.

3.2.1. Question Generation

Unlike previous approaches that rely on well-annotated data [2, 70] or learnable question generation [49], we generate three types of questions from ground truth captions: **What**, **Why**, and **How**. **What** questions target concrete details (e.g., *"What does the person pick up while putting away the dishes?"*), improving visual recognition. **Why** questions focus on causal reasoning (e.g., *"Why does the person put the dishes in the cabinet?"*), enhancing intent understanding. **How** questions drive procedural understanding (e.g., *"How does the person clean the floor using the vacuum cleaner?"*), strengthening temporal reasoning. Questions are regenerated in each self-training iteration, though *Why* questions may occasionally induce hallucination.

3.2.2. Answer Generation

For each generated question, the model produces five candidate answers at different temperatures ([0.3, 0.5, 0.7, 0.9, 1.0]). Using different temperature values helps ensure diversity in the generated responses while maintaining varying degrees of creativity and precision. Lower temperatures (i.e., 0.3, 0.5) tend to produce more focused and conservative responses, while higher temperatures (i.e., 0.7, 1.0) allow for more diverse and creative answers.

3.2.3. Preference Selection

This component of our framework is critical, as it evaluates the five answers generated in the previous stage for each of the three previously derived questions. Unlike prior work that relies on paid APIs [70] or explicit visual context for preference selection [1, 2], our approach enables the Video-LLM to serve as its own evaluator. Inspired by recent self-rewarding works [11, 40, 65], the model assesses its responses using only the video caption as reference. This self-evaluation process facilitates the creation of high-quality preference data, which is then used to fine-tune the model via DPO. The evaluation criteria include five key metrics to ensure that the model's responses are contextually relevant and grounded in the video:

1. **Relevance**: Assesses how well the generated answer addresses the question, ensuring logical coherence between question and answer.
2. **Accuracy**: Measures the factual correctness of the answer based on the video caption, crucial for avoiding hallucinations.
3. **Temporal Grounding**: Evaluates whether the answer correctly references the timing and sequence of events, essential for time-sensitive responses in video tasks.
4. **Clarity**: Ensures that the answer is clear, concise, and free from grammatical errors, enhancing usability and accessibility.
5. **Groundedness**: Checks if the answer relies solely on information from the video caption without introducing unsupported details, maintaining factual consistency.

By applying these criteria, the model ranks its responses from 1 (lowest) to 5 (highest).

3.2.4. CLIP Filtering of Preference Data

To enhance the robustness of our preference data for DPO, we address instances where the model's self-evaluation may incorrectly rank generated responses in initial iterations of self-training. Specifically, there are cases where the rejected answer is more aligned with the video content than the chosen answer. To mitigate this issue, we introduce a CLIP-based [42] adjustment mechanism that leverages CLIP's ability to measure the similarity between images (video frames) and text. For each pair of positive and negative examples (a_k^+, a_k^-), we compute the CLIP similarity scores between the video v_j and each answer:

$$c_k^+ = \text{CLIP}(v_j, a_k^+), \quad c_k^- = \text{CLIP}(v_j, a_k^-) \tag{1}$$

Based on these scores, we determine a per-example sign s_k to adjust the DPO loss:

$$s_k = \begin{cases} +1, & \text{if } c_k^+ \geq c_k^- \\ -1, & \text{if } c_k^+ < c_k^- \end{cases} \tag{2}$$

This adjustment ensures that if CLIP indicates the negative example is more aligned with the video content, the loss function is modified to favor the negative example during training. The adjusted DPO loss is defined as:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\log \sigma \left(\beta s_k \left(\delta_\theta - \delta_{\text{ref}} \right) \right), \tag{3}$$

where $\delta_\theta = \log p_\theta(a_k^+ \mid q_k, v_j) - \log p_\theta(a_k^- \mid q_k, v_j)$, $\delta_{\text{ref}} = \log p_{\text{ref}}(a_k^+ \mid q_k, v_j) - \log p_{\text{ref}}(a_k^- \mid q_k, v_j)$, β is a scaling factor, and $\sigma(\cdot)$ is the sigmoid function. This per-example adjustment allows the training process to favor responses that are better aligned with the video content, as determined by CLIP, thereby improving the model's alignment between vision and text.

3.2.5. DPO Finetuning

After CLIP filtering, we fine-tune using DPO with scaling factors $\alpha = 1.0$ and $\beta = 0.1$. Following [12], we incorporate regularization through log probability ratios between current and reference models:

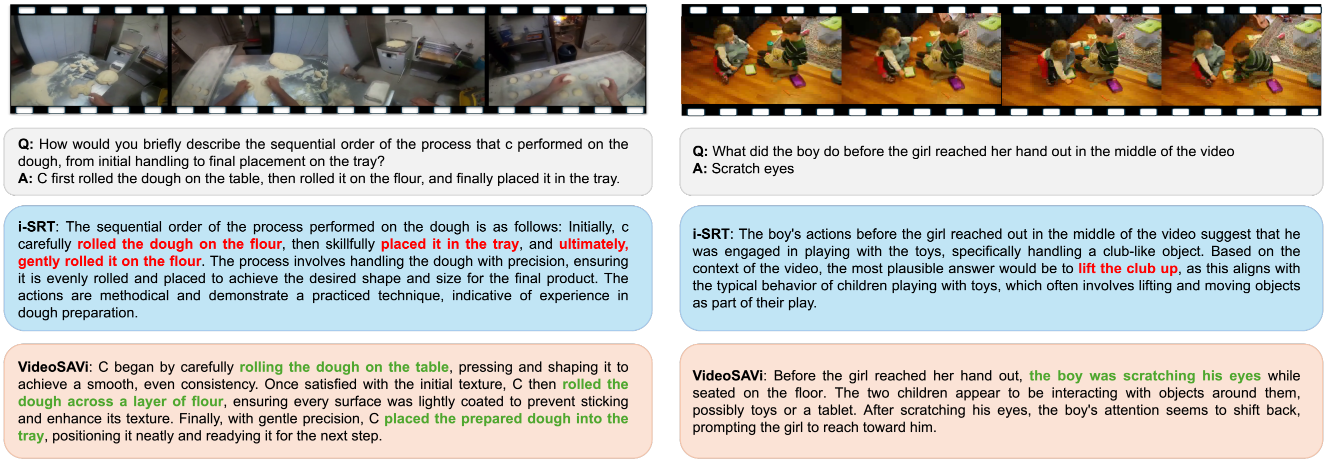
$$\mathcal{L}_{\text{Total}}(\theta) = -\sum_{\mathcal{D}} \Big[\log \sigma \left(\beta s_k (\delta_\theta - \delta_{\text{ref}}) \right) + \lambda \log p_\theta(a_k^+ \mid v_j, q_k) \Big] \tag{4}$$

The CLIP-adjusted sign s_k modifies the preference gradient direction, ensuring that the model learns to favor responses better aligned with visual content. Specifically, when $s_k = -1$, the gradient encourages decreasing the probability ratio between the originally chosen response and the alternative, effectively inverting the preference based on CLIP's visual similarity assessment. This adjustment helps prevent the model from overfitting to purely language-based preferences and maintains visual grounding throughout the optimization process. Additionally, the regularization term $\lambda \log p_\theta(a_k^+ \mid v_j, q_k)$ prevents excessive deviation from the reference model's learned knowledge while allowing sufficient flexibility to incorporate new video-text alignments.

While we investigated DPO variants such as Robust-DPO [44], IPO [3], and hinge-based loss [32], standard sigmoid-based DPO offered the optimal balance between performance and efficiency. By refining preference gradients to favor visually aligned, semantically appropriate responses, our approach uses DPO to align model outputs with video content—improving video-text grounding without requiring external labeled data or proprietary models.

4. Experiments and Results

To evaluate the effectiveness of our proposed method, we conduct extensive experiments and ablation studies across various benchmarks, comparing our results with SOTA models in the literature.



Q: How would you briefly describe the sequential order of the process that c performed on the dough, from initial handling to final placement on the tray?
A: C first rolled the dough on the table, then rolled it on the flour, and finally placed it in the tray.

i-SRT: The sequential order of the process performed on the dough is as follows: Initially, c carefully **rolled the dough on the flour**, then skillfully **placed it in the tray**, and **ultimately, gently rolled it on the flour**. The process involves handling the dough with precision, ensuring it is evenly rolled and placed to achieve the desired shape and size for the final product. The actions are methodical and demonstrate a practiced technique, indicative of experience in dough preparation.

VideoSAVi: C began by carefully **rolling the dough on the table**, pressing and shaping it to achieve a smooth, even consistency. Once satisfied with the initial texture, C then **rolled the dough across a layer of flour**, ensuring every surface was lightly coated to prevent sticking and enhance its texture. Finally, with gentle precision, C **placed the prepared dough into the tray**, positioning it neatly and readying it for the next step.

Q: What did the boy do before the girl reached her hand out in the middle of the video
A: Scratch eyes

i-SRT: The boy's actions before the girl reached out in the middle of the video suggest that he was engaged in playing with the toys, specifically handling a club-like object. Based on the context of the video, the most plausible answer would be to **lift the club up**, as this aligns with the typical behavior of children playing with toys, which often involves lifting and moving objects as part of their play.

VideoSAVi: Before the girl reached her hand out, **the boy was scratching his eyes** while seated on the floor. The two children appear to be interacting with objects around them, possibly toys or a tablet. After scratching his eyes, the boy's attention seems to shift back, prompting the girl to reach toward him.

Figure 3. **Qualitative examples showing VideoSAVi's fine-grained video understanding capabilities.** Left: Our model provides a precise temporal sequence for dough preparation ("rolling on table" → "rolling across flour" → "placing in tray"), in contrast to i-SRT's [1] imprecise ordering of similar actions. Right: VideoSAVi accurately captures subtle behavioral details such as "scratching eyes" that occurred before the girl's interaction, while baseline i-SRT incorrectly predicts generic toy-playing actions.

Dataset	Instruction Pairs	#Vid	Total Questions
LLaVA [31]	665k	259k*	–
VInstruct [35]	100k	13.3k	–
Self-Training Data			
Star [58]	–	15,970	23,955
Vidal [73]	–	3,994	5,991
WebVid [5]	–	3,970	5,955
Generated Pref.	–	11.9k	35.9k
Total	765k	308k†	35.9k

Table 1. **Dataset statistics for VideoSAVi training.** *Images only. †Includes images and videos.

4.1. Experimental Setup

Model Architecture. We train two variants of VideoSAVi. The first variant builds on VideoLLaVA [29], using pre-trained weights from LLaVA-HOUND [70]. It integrates Vicuna-7B-v1.5 [72] as the LLM, with LanguageBind [73] as the image and video encoder. Additionally, we incorporate a LoRA adapter [16] for efficient fine-tuning. The second variant of VideoSAVi adopts the LLaVA-NeXT architecture [20] with interleaving [21]. This model uses Qwen v1.5 [4] as the LLM, which we evaluate in both its 0.5B and 7B versions. For visual encoding, it employs the SigLIP-400M model [67]. This variant is initialized directly with SFT weights, thus skipping the warmup SFT fine-tuning phase.

Training Datasets. For Stage I (supervised fine-tuning, we perform warmup training using the VideoInstruct-100K [35] and LLaVA instruction tuning datasets [31]. For Stage II (self-training), we use raw video data from the STAR (Charades) [58], WebVid [5], and Vidal [73] datasets, from which our model generates preference data. Table 1 provides a detailed breakdown of all the datasets used.

4.2. Benchmarks

We evaluate VideoSAVi on the following tasks:

1. **Temporal Reasoning:** For this task, we use the TempCompass benchmark [33] for evaluation, which covers various temporal aspects such as action (identification of coarse and fine-grained movements), attribute change (recognition of temporal changes in object properties like size, shape, and color), speed (perception of absolute and relative motion velocity), direction (comprehension of object and camera movement orientation), and event order (understanding the chronological sequence of events in videos) across four main tasks: (1) multi-choice QA, (2) yes/no QA, (3) caption matching, and (4) caption generation. The evaluation is conducted using GPT-3.5-Turbo-0125.
2. **Multi-choice QA:** For this task, the model must choose the correct answer from a given set of choices. The benchmarks include NExTQA [59], EgoSchema [36] and IntentQA [24].
3. **Open-ended QA:** For this task, the model generates answers to questions related to the video in freestyle form without any choices provided in the input prompt. This is a zero-shot QA evaluation performed on three datasets: MSVD-QA [8], MSRVTT-QA [60], and TGIF-QA [27]. This evaluation is also conducted using GPT-3.5-Turbo-0125.

Method	LLM	Act	Dir	Spd	Evt	Attr	Avg
SF-LLaVA [61]	7B	52.3	33.5	27.4	36.3	33.1	36.5
Video-ChatGPT [35]	7B	51.3	39.7	37.6	42.3	40.9	42.4
Video-LLaMA [68]	13B	62.4	34.8	34.5	43.8	40.4	43.2
LLaMA-VID [28]	7B	61.8	38.1	38.2	55.9	40.9	47.0
Video-LLAVA [29]	7B	70.9	41.6	43.4	46.5	38.3	48.1
Video-STaR [74]	7B	81.4	36.9	38.2	37.1	**50.7**	50.3
VideoChat2 [26]	7B	70.1	**43.8**	45.3	44.8	48.6	50.5
VideoSAVi-Vicuna	7B	67.7	41.4	39.4	40.3	49.9	47.7
VideoSAVi-Qwen	.5B	64.2	42.7	44.7	44.5	47.9	48.8
VideoSAVi-Qwen	7B	**83.2**	42.3	**48.4**	**46.9**	49.6	**54.1**
Proprietary Model Preference Data							
LLaVA-NeXT [30]	7B	79.6	45.2	44.3	48.0	47.7	53.0
LLaVA-HOUND [70]	7B	80.4	44.1	43.5	50.6	58.9	55.5
i-SRT [1]	7B	80.6	43.4	44.7	52.4	58.7	56.0

Table 2. Temporal reasoning results on TempCompass [33]. All values are in percentages (%). Act: Action (activity recognition), Dir: Direction (motion tracking), Spd: Speed (motion rate), Evt: Event (sequence ordering), and Attr: Attribute (property changes).

Method	LLM	NExTQA	EgoQA	IntentQA
Video-LLaMA2 [10]	7B	–	51.7	–
MovieChat+ [47]	7B	54.8	–	–
Vista-LLM [34]	7B	60.7	–	–
Video-LLAVA [29]	7B	49.2	18.8	48.6
SF-LLaVA [61]	7B	64.2	47.1	60.1
VideoAgent [54]	GPT4	71.3	60.2	–
VideoTree [56]	GPT4	73.5	**66.2**	66.9
VideoSAVi-Vicuna	7B	38.2	10.2	41.6
VideoSAVi-Qwen	0.5B	57.8	31.6	54.7
VideoSAVi-Qwen	7B	**74.1**	57.2	**76.4**
Proprietary Model Preference Data				
LLaVA-NeXT [30]	7B	52.4	35.0	53.5
LLaVA-HOUND [70]	7B	61.6	36.1	58.6
i-SRT [1]	7B	63.0	46.2	59.3

Table 3. Multi-choice QA results on NExTQA [59], EgoSchema (EgoQA) [36] and IntentQA [24] benchmarks. All values are in percentages (%).

4.3. Results

We compare VideoSAVi with various SOTA video language models. Our main comparison is with another self-training-based model called Video-STaR [74], as well as methods that use proprietary models to generate DPO preference data, such as LLaVA-NeXT-DPO [30], LLaVA-HOUND-DPO [70], and i-SRT [1]. We also compare our approach with training-free methods such as VideoAgent [54], VideoTree [56], and Slow-Fast-LLaVA [61]. Tables 2, 3, and 4 summarize the results for the tasks of temporal reasoning, multi-choice QA, and open-ended QA, respectively.

Method	LLM	MSVD		MSRVTT		TGIF	
		Acc	Scr	Acc	Scr	Acc	Scr
Video-ChatGPT [35]	7B	68.6	3.8	58.9	3.4	47.8	3.2
LLaMA-VID [28]	7B	69.7	3.7	57.5	3.2	–	–
Video-LLAVA [29]	7B	70.1	3.9	58.6	3.5	49.1	3.0
VideoChat2 [26]	7B	70.0	3.9	54.1	3.3	–	–
Video-STaR [74]	7B	71.3	4.0	58.2	**3.5**	47.3	3.3
VLM-RLAIF [2]	7B	75.1	3.9	**61.0**	3.3	–	–
VideoSAVi-Qwen	.5B	59.0	3.4	44.1	2.9	41.3	2.9
VideoSAVi-Qwen	7B	69.2	3.8	54.0	3.2	49.4	3.2
VideoSAVi-Vicuna	7B	**76.0**	**4.1**	60.1	**3.5**	**53.1**	**3.4**
Proprietary Model Preference Data							
LLaVA-NeXT [30]	7B	75.4	4.0	62.4	3.5	54.4	3.4
LLaVA-HOUND [70]	7B	78.7	4.0	69.0	3.7	60.7	3.5
i-SRT [1]	7B	81.3	4.1	72.8	3.8	62.0	3.5

Table 4. Open-ended QA results on MSVD [8], MSRVTT [60], and TGIF [27]. VideoSAVi-Vicuna achieves the best performance among open-source models. All values are in percentages (%). Acc: Accuracy, Scr: GPT-evaluated quality score (1–5).

Temporal Reasoning VideoSAVi shows significant improvements in temporal reasoning on TempCompass dataset, excelling in both coarse-grained (e.g., running, cooking) and fine-grained (e.g., climbing up/down a ladder, specific sports movements such as dribbling or dunking a basketball) action recognition. For instance, VideoSAVi-Qwen-7B achieves 83.2% accuracy on action understanding, surpassing the previous SOTA methods, such as Video-STaR (81.4%) and i-SRT (80.6%). This suggests that VideoSAVi's use of "What" questions enhances its ability to recognize and understand specific actions within videos. In speed perception, which includes absolute speed assessment (e.g., detecting slow motion vs. normal speed) and relative speed comparison (e.g., comparing velocities of different objects within the same frame), VideoSAVi reaches 48.4% accuracy, improving upon the previous baseline of i-SRT (44.7%). This is significant since speed perception requires strong temporal reasoning involving both motion detection and accurate assessment of temporal rates both within and across frames. Additionally, VideoSAVi's smaller 0.5B model (VideoSAVi-Qwen-0.5B) achieves an impressive efficiency-performance trade-off, maintaining 64.2% accuracy in action recognition and 44.7% in speed assessment while using far fewer parameters than existing methods and outperforming 7B models in overall accuracy.

Multi-choice QA VideoSAVi-Qwen-7B achieves new SOTA performance on NExTQA (74.1%) and IntentQA (76.4%), surpassing both proprietary and GPT4-based models. On the long-video understanding task of EgoSchema, it reaches 57.2%, competitive with most models, except for VideoTree's 66.2% (GPT-4 based). VideoSAVi-Qwen-0.5B also maintains strong performance, with 57.8% accuracy on NExTQA and 54.7% on IntentQA. These results corroborate with the inclusion of "Why" questions in our question generation process which significantly improves the model's intent understanding capabilities. VideoSAVi-Vicuna, however, demonstrates lower accuracy due to its reduced instruction following ability in choosing correct choice out of given answer choices.

Open-ended QA VideoSAVi-Vicuna-7B achieves strong performance on MSVD-QA (76.0%) surpassing LLaVA-NeXT-DPO (75.4%) which uses proprietary models to generate preference data while also maintaining strong results on MSRVTT-QA (60.1%) and TGIF-QA (53.1%). VideoSAVi-Vicuna-0.5B also retains 77.6% of the 7B model's MSVD-QA accuracy. VideoSAVi-Vicuna-7B also achieves the highest scores (4.1, 3.5, 3.4) in all three benchmarks. However, like LLaVA-HOUND-DPO [70] and i-SRT [1], the Vicuna variant produces verbose outputs for this benchmark, and GPT-based evaluations tend to assign higher scores to such responses [41].

5. Ablation Studies

We conduct three ablation studies on the TempCompass benchmark to examine the impact of (1) removing CLIP filtering, (2) removing caption references during question generation and evaluation, and (3) removing "How" and "Why" questions. Table 5 shows the results of these studies.

5.1. Effect of Removing CLIP Filtering

We remove CLIP filtering of preference data and re-evaluate the models on TempCompass. Removing CLIP filtering significantly degrades performance: VideoSAVi-Vicuna-7B drops from an average of 47.7% to 28.7%, VideoSAVi-Qwen-0.5B from 48.8% to 43.6% (with a directional understanding drop from 42.7% to 38.4%), and VideoSAVi-Qwen-7B from 54.1% to 49.5%. These results confirm the importance of CLIP filtering for maintaining video-text alignment and accurate response ranking.

5.2. Effect of Removing Captions

Removing caption references during question generation and evaluation has varying impacts across models. VideoSAVi-Vicuna-7B's performance drops significantly from an average of 47.7% to 33.6%, with major declines in action recognition (67.7% → 48.2%) and direction understanding (41.4% → 27.4%). However, the VideoSAVi-Qwen models show strong robustness: the 7B variant maintains similar performance (54.3% vs. 54.1%), and the 0.5B model shows only a minimal decline (47.7% vs. 48.8%), suggesting that these models can effectively reason about temporal aspects even without caption guidance.

5.3. Effect of Removing "How" and "Why"

When restricted to only "What" questions, VideoSAVi-Vicuna-7B's performance drops from an average of 47.7% to 44.6%. VideoSAVi-Qwen-0.5B shows minimal degradation decreasing from 48.8% to 47.7%. Interestingly, VideoSAVi-Qwen-7B maintains similar performance (54.2% vs 54.1%), suggesting that "How" and "Why" questions had little effect on the 7B model.

Method	LLM	Act	Dir	Spd	Evt	Attr	Avg
CLIP Filtering Removed							
VideoSAVi-Vicuna	7B	51.7	22.7	17.6	22.6	29.1	28.7
VideoSAVi-Qwen	.5B	58.7	38.4	37.8	41.1	42.1	43.6
VideoSAVi-Qwen	7B	75.5	38.6	42.7	45.1	45.8	49.5
Caption Reference Removed							
VideoSAVi-Vicuna	7B	48.2	27.4	26.0	29.0	37.2	33.6
VideoSAVi-Qwen	.5B	64.3	42.1	43.0	44.0	45.4	47.7
VideoSAVi-Qwen	7B	83.5	43.1	48.5	45.8	50.8	54.3
Only "What" Questions							
VideoSAVi-Vicuna	7B	61.7	39.2	39.0	34.3	48.6	44.6
VideoSAVi-Qwen	.5B	63.5	41.8	44.1	42.8	46.4	47.7
VideoSAVi-Qwen	7B	83.2	43.0	47.9	46.7	50.4	54.2

Table 5. **Ablation study on TempCompass benchmark.** We examined the impact of (1) removing CLIP filtering, (2) removing caption references during question generation and evaluation, and (3) removing "How" and "Why" questions. Results are reported across action (Act), direction (Dir), speed (Spd), event (Evt), and attribute (Attr) understanding tasks.

6. Limitations

While our approach shows considerable promise, it also presents some areas for improvement: (1) computational demands: experiments required 4 Nvidia A100 GPUs (80GB each), with one iteration of self-training taking approximately 5 days; (2) instruction adherence: model responses tend to become more verbose during DPO alignment, suggesting a balance to be refined between video-text alignment and conciseness; and (3) synthetic data quality: self-generated preferences may occasionally diverge from human preferences in complex reasoning scenarios. Future work will focus on enhancing computational efficiency and refining the balance between visual alignment and instruction-following capabilities.

7. Conclusion

We present VideoSAVi, a novel self-training framework that enables video-language models to generate and learn from their own synthetic preference data without relying on human annotations or proprietary models. Through CLIP-adjusted DPO and iterative self-improvement, VideoSAVi achieves SOTA performance on temporal reasoning and video QA tasks. Our approach demonstrates that even smaller models can effectively generate high-quality training data, opening new possibilities for democratizing the development of Video-LLMs while maintaining strong performance.

Acknowledgments

This research was supported by the National Eye Institute (NEI) of the National Institutes of Health (NIH) under award number R01EY034562. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

[1] Daechul Ahn, Yura Choi, San Kim, Youngjae Yu, Dongyeop Kang, and Jonghyun Choi. i-srt: Aligning large multimodal models for videos by iterative self-retrospective judgment. *arXiv preprint arXiv:2406.11280*, 2024. 2, 3, 5, 6, 7, 8, 17, 18

[2] Daechul Ahn, Yura Choi, Youngjae Yu, Dongyeop Kang, and Jonghyun Choi. Tuning Large Multimodal Models for Videos using Reinforcement Learning from AI Feedback. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 923–940, 2024. 2, 3, 4, 5, 7

[3] Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, and et al. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 4447–4455, 2024. 5

[4] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, and et al. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023. 6

[5] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1728–1738, 2021. 4, 6

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, and Jared D Kaplan. Language Models are Few-Shot Learners. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 1877–1901. Curran Associates, Inc., 2020. 1

[7] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, and et al. Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback. *Transactions on Machine Learning Research (TMLR)*, 2023. 2

[8] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 190–200, 2011. 6, 7

[9] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, and et al. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18407–18418, 2024. 2

[10] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, and et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 7

[11] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, and et al. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 9722–9744, 2024. 5

[12] Yihe Deng, Pan Lu, Fan Yin, Ziniu Hu, Sheng Shen, James Zou, Kai-Wei Chang, and Wei Wang. Enhancing large vision language models with self-training on image comprehension. *arXiv preprint arXiv:2405.19716*, 2024. 2, 3, 5

[13] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Letman, and et al. The Llama 3 Herd of Models, 2024. 1

[14] Hiroki Furuta, Kuang-Huei Lee, Shixiang Shane Gu, Yutaka Matsuo, Aleksandra Faust, Heiga Zen, and Izzeddin Gur. Geometric-averaged preference optimization for soft preference labels. *arXiv preprint arXiv:2409.06691*, 2024. 2

[15] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, and Lotte et al. Weerts. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023. 3

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 6

[17] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14271–14280, 2024. 2

[18] Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large Language Models Can Self-Improve. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1051–1068, 2023. 3

[19] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 26874–26901, 2024. 2

[20] Bo Li, Kaichen Zhang, Hao Zhang, Dong Guo, Renrui Zhang, Feng Li, Yuanhan Zhang, Ziwei Liu, and Chunyuan Li. Llava-next: Stronger llms supercharge multimodal capabilities in the wild, 2024. 6

[21] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 6

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

[23] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 19730–19742, 2023. 1

[24] Jiapeng Li, Ping Wei, Wenjuan Han, and Lifeng Fan. Intentqa: Context-aware video intent reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11963–11974, 2023. 6, 7, 17

[25] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 1

[26] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22195–22206, 2024. 7

[27] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. Tgif: A new dataset and benchmark on animated gif description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4641–4650, 2016. 6, 7

[28] 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 (ECCV)*, pages 323–340. Springer, 2025. 7

[29] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 1, 2, 6, 7

[30] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 3, 7, 17, 18

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2024. 1, 3, 6

[32] Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. 5

[33] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. TempCompass: Do video LLMs really understand videos? In *Findings of the Association for Computational Linguistics (ACL)*, pages 8731–8772. Association for Computational Linguistics, 2024. 6, 7, 15

[34] Fan Ma, Xiaojie Jin, Heng Wang, Yuchen Xian, Jiashi Feng, and Yi Yang. Vista-llama: Reducing Hallucination in Video Language Models via Equal Distance to Visual Tokens . In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13151–13160, 2024. 7

[35] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 1, 3, 6, 7

[36] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. EgoSchema: A diagnostic benchmark for very long-form video language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 46212–46244. Curran Associates, Inc., 2023. 6, 7

[37] OpenAI. GPT-4V(ision) system card, 2024. 2

[38] OpenAI. GPT-4 Technical Report, 2024. 1, 3

[39] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, and Zhang. Training language models to follow instructions with human feedback. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 35:27730–27744, 2022. 1, 2

[40] Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *arXiv preprint arXiv:2404.19733*, 2024. 5

[41] Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. In *Findings of the Association for Computational Linguistics (ACL)*, pages 4998–5017, 2024. 8, 14

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, and Sandhini Agarwal. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 1, 3, 5, 12

[43] 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. In *Thirty-Seventh Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 2, 3, 12

[44] Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust DPO: Aligning language models with noisy feedback. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 42258–42274, 2024. 5

[45] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 2

[46] Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, and et al Garcia. Beyond Human Data: Scaling Self-Training for Problem- Solving with Language Models. *Transactions on Machine Learning Research (TMLR)*, 2024. 3

[47] Enxin Song, Wenhao Chai, Tian Ye, Jenq-Neng Hwang, Xi Li, and Gaoang Wang. Moviechat+: Question-aware sparse memory for long video question answering. *arXiv preprint arXiv:2404.17176*, 2024. 7

[48] Guohao Sun, Can Qin, Huazhu Fu, Linwei Wang, and Zhiqiang Tao. Stllava-med: Self-training large language and vision assistant for medical. *arXiv preprint arXiv:2406.19973*, 2024. 3

[49] Guohao Sun, Can Qin, Jiamian Wang, Zeyuan Chen, Ran Xu, and Zhiqiang Tao. Sq-llava: Self-questioning for large vision-language assistant. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 156–172, 2025. 2, 3, 4

[50] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, and et al. Aligning Large Multimodal Models with Factually Augmented RLHF. In *Findings of the Association for Computational Linguistics (ACL)*, pages 13088–13110, 2024. 2

[51] Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Daniel Cox, Yiming Yang, and Chuang Gan. SALMON: Self-Alignment with Instructable Reward Models. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. 2

[52] Reuben Tan, Ximeng Sun, Ping Hu, Jui-hsien Wang, Hanieh Deilamsalehy, and et al. Koala: Key frame-conditioned long video-llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13581–13591, 2024. 2

[53] Tianduo Wang, Shichen Li, and Wei Lu. Self-Training with Direct Preference Optimization Improves Chain-of-Thought Reasoning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 11917–11928, 2024. 3

[54] Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. *arXiv preprint arXiv:2403.10517*, 2024. 7

[55] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13484–13508. Association for Computational Linguistics, 2023. 3

[56] Ziyang Wang, Shoubin Yu, Elias Stengel-Eskin, Jaehong Yoon, Feng Cheng, Gedas Bertasius, and Mohit Bansal. Videotree: Adaptive tree-based video representation for llm reasoning on long videos. *arXiv preprint arXiv:2405.19209*, 2024. 7

[57] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned Language Models are Zero-Shot Learners. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 1

[58] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. Star: A benchmark for situated reasoning in real-world videos. In *Proceedings of the Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*, 2021. 4, 6

[59] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9777–9786, 2021. 6, 7, 18

[60] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5288–5296, 2016. 6, 7

[61] Mingze Xu, Mingfei Gao, Zhe Gan, Hong-You Chen, Zhengfeng Lai, Haiming Gang, Kai Kang, and Afshin Dehghan. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*, 2024. 7

[62] Zhiyang Xu, Ying Shen, and Lifu Huang. MultiInstruct: Improving Multi-Modal Zero-Shot Learning via Instruction Tuning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 11445–11465, 2023. 1

[63] Wei Jie Yeo, Teddy Ferdinan, Przemyslaw Kazienko, Ranjan Satapathy, and Erik Cambria. Self-training large language models through knowledge detection. *arXiv preprint arXiv:2406.11275*, 2024. 3

[64] Yueqin Yin, Zhendong Wang, Yujia Xie, Weizhu Chen, and Mingyuan Zhou. Self-augmented preference optimization: Off-policy paradigms for language model alignment. *arXiv preprint arXiv:2405.20830*, 2024. 3

[65] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 57905–57923, 2024. 5

[66] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STaR: Bootstrapping Reasoning With Reasoning. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 35:15476–15488, 2022. 3

[67] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid Loss for Language Image Pre-Training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952, 2023. 6, 15

[68] Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 543–553. Association for Computational Linguistics, 2023. 2, 7

[69] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 543–553, 2023. 1

[70] Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu, Chunyuan Li, Alexander Hauptmann, Yonatan Bisk, et al. Direct preference optimization of video large multimodal models from language model reward. *arXiv preprint arXiv:2404.01258*, 2024. 3, 4, 5, 6, 7, 8, 17, 18

[71] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. Llava-next: A strong zero-shot video understanding model, 2024. 2, 3

[72] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, and et al. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 46595–46623, 2023. 6

[73] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, and et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *Proceedings of the The International Conference on Learning Representations (ICLR)*, 2024. 4, 6

[74] Orr Zohar, Xiaohan Wang, Yonatan Bitton, Idan Szpektor, and Serena Yeung-Levy. Video-star: Self-training enables video instruction tuning with any supervision. *arXiv preprint arXiv:2407.06189*, 2024. 3, 7

A. Analysis of CLIP-Adjusted DPO

We provide a detailed analysis of the proposed CLIP-adjusted direct preference optimization (DPO) method. We aim to show the challenges associated with the introduction of CLIP-based [42] adjustments on the model's behavior, leading to increased response verbosity, decreased instruction following, and higher supervised fine-tuning (SFT) loss.

A.1. Base DPO Loss Formulation

The standard DPO loss [43] is formulated to optimize a policy model $p_\theta(y|x)$ by encouraging it to prefer responses that are deemed better according to human or model-generated preferences. The loss is defined as:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\,\mathbb{E}_{(x,\,y^+,\,y^-)\sim\mathcal{D}}\left[\log\sigma\left(\beta\,\Delta r_\theta\right)\right], \tag{5}$$

where $\Delta r_\theta = r_\theta(x, y^+) - r_\theta(x, y^-)$, $r_\theta(x, y) = \log\frac{p_\theta(y|x)}{p_{\text{ref}}(y|x)}$, $\sigma(\cdot)$ is the sigmoid function, $\beta > 0$ is a scaling parameter, x is the input (video and prompt), and (y^+, y^-) are the chosen and rejected responses from dataset $\mathcal{D}$.

The DPO loss encourages the policy model to assign higher probability to the preferred response compared to the less preferred one relative to the reference model.

A.2. CLIP-Adjusted DPO Loss

In our CLIP-adjusted DPO, we introduce a per-example sign coefficient s_i based on the CLIP similarity scores between the responses and the video content. Specifically, for each example i, we compute:

$$s_i = \begin{cases} +1, & \text{if } c_i^+ \geq c_i^-, \\ -1, & \text{if } c_i^+ < c_i^-, \end{cases} \tag{6}$$

where

- $c_i^+ = \text{CLIP}(v_i, y_i^+)$ is the CLIP similarity score between the video v_i and the preferred response y_i^+.
- $c_i^- = \text{CLIP}(v_i, y_i^-)$ is the CLIP similarity score between the video v_i and the less preferred response y_i^-.

The adjusted DPO loss becomes:

$$\mathcal{L}_{\text{CLIP-DPO}}(\theta) = -\,\mathbb{E}_{(x,\,y^+,\,y^-)\sim\mathcal{D}}\left[\log\sigma\left(\beta\,s_i\,\Delta r_\theta\right)\right]. \tag{7}$$

This adjustment effectively flips the preference direction in cases where the CLIP similarity suggests that the less preferred response is more aligned with the video content than the preferred response.

A.3. Combined Loss Function with SFT Regularization

To balance the optimization between aligning responses with visual content and maintaining language modeling capabilities, we introduce a combined loss function that includes an SFT loss term:

$$\mathcal{L}_{\text{Total}}(\theta) = \alpha\,\mathcal{L}_{\text{CLIP-DPO}}(\theta) + \gamma\,\mathcal{L}_{\text{SFT}}(\theta), \tag{8}$$

where

- $\alpha > 0$ and $\gamma > 0$ are weighting factors for the CLIP-DPO loss and SFT loss, respectively.
- $\mathcal{L}_{\text{SFT}}(\theta)$ is the SFT loss, defined as the negative log-likelihood of the ground-truth tokens:

  $$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x,\,y)\sim\mathcal{D}_{\text{SFT}}}\left[\log p_\theta(y|x)\right], \tag{9}$$

  where $\mathcal{D}_{\text{SFT}}$ is the SFT dataset containing video-response (chosen) pairs. We include $\mathcal{L}_{\text{SFT}}$ as a regularization term to prevent the model from overly optimizing for CLIP-based preferences at the cost of coherent language generation, since DPO alone might lead to unnatural or verbose outputs as shown in Figure 4.

A.4. Impact on Model Behavior

The introduction of the CLIP-adjusted DPO loss and its interaction with the SFT loss have significant implications for the model's behavior. We analyze these effects below.

A.4.1. Gradient Analysis

The gradient of the CLIP-adjusted DPO loss with respect to the model parameters θ is given by:

$$\nabla_\theta\mathcal{L}_{\text{CLIP-DPO}} = -\beta\,\mathbb{E}_{\mathcal{D}}\Big[s_i\left(1-\sigma(\beta\,s_i\,\Delta r_\theta)\right) \times \nabla_\theta\Delta r_\theta\Big], \tag{10}$$

where $s_i = -1$, the sign of the gradient is reversed compared to the standard DPO loss. This reversal can cause the model to decrease the difference Δr_θ, effectively encouraging the model to assign higher probability to the less preferred response y^- relative to the preferred response y^+. This conflicting signal introduces instability in the optimization process.

Effect on Gradient Magnitude. The magnitude of the gradient is modulated by the term $(1-\sigma\left(\beta\,s_i\,\Delta r_\theta\right))$. This term can vary significantly depending on the value of $\beta\,s_i\,\Delta r_\theta$. For large values of $|\beta\,s_i\,\Delta r_\theta|$, the sigmoid function saturates, leading to vanishing gradients. Conversely, for small values, the gradient magnitude increases, potentially causing abrupt updates.

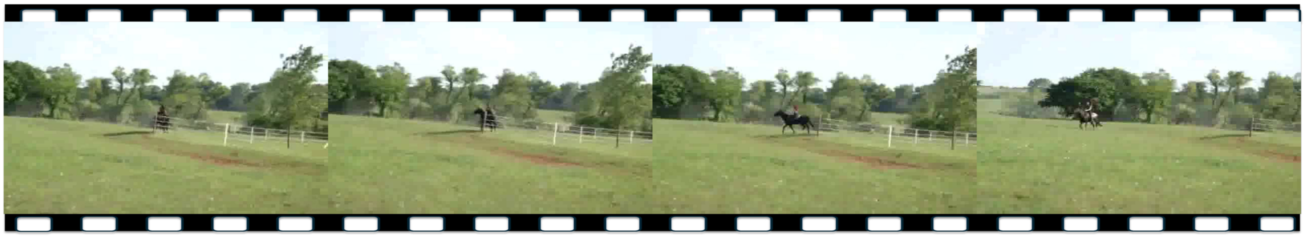
A.4.2. Increase in Response Verbosity

The reversal of gradients for examples where $s_i = -1$ can lead the model to hedge its predictions. To accommodate conflicting optimization signals, the model may increase the entropy of its output distribution, resulting in more verbose and less precise responses.

Entropy Increase. The entropy of the policy model's output distribution is defined as:

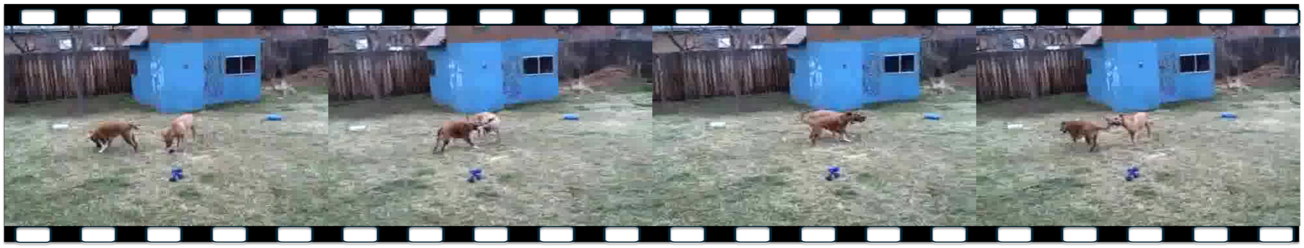
$$H\left(p_\theta(y|x)\right) = -\sum_{y} p_\theta(y|x)\log p_\theta(y|x), \tag{11}$$

where $p_\theta(y|x)$ is the conditional probability distribution of the policy model generating response y given input x. An



Q: Why did the horse change direction after running for a while?
Options. A. change posture to support baby B. to prepare to push harder C. reaches the fence D. wave push him
E. shot the gun
Ans. C

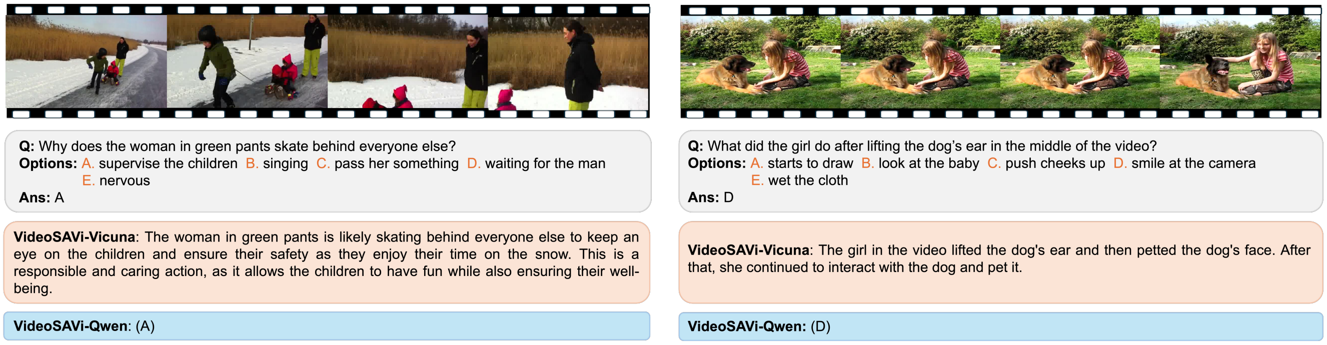
The horse changed direction after running for a while, likely due to a command from the rider or as part of a training exercise. The options provided suggest various reasons for this change in direction. The most plausible explanation, given the context of horse riding and training, is (B) to prepare to push harder. This option implies that the horse is being conditioned or exercised to increase its speed or endurance, and changing direction is a common training technique to prepare the horse for more strenuous activity. Options (A), (C), (D), and (E) are less likely as they do not align with common equestrian training practices. Option (A) is not supported by the context, as horses do not change posture to support a baby. Option (C) is incorrect because the horse did not reach a fence, which is a typical obstacle in equestrian sports. Option (D) is also unlikely as there is no evidence of a wave pushing the horse. Lastly, option (E) is not applicable as there is no indication of a gun being used in the training session. Therefore, **the most logical answer is (B) to prepare to push harder**.



Q: Why are the dogs running after each other?
Options. A. try to bite each other B. follow the ball C. chasing the car D. chase the bottle E. guide direction
Ans. D

The dogs are displaying classic playful behavior in this backyard scene, running and interacting with clear signs of friendly engagement. Looking at their relaxed body language and coordinated movements, we can see they're focused on what appears to be a blue ball on the ground. Dogs naturally love to chase and play with balls - it's a fundamental part of their social and recreational behavior. When we examine the options (A through E), **option B 'follow the ball' stands out as the most logical choice**. We can rule out aggressive behavior (option A) since there's no sign of hostility, and options C, D, and E don't match what we observe - there's no car or bottle present, and the dogs aren't engaged in any guiding behavior. Their synchronized running and mutual focus strongly suggest they're simply playing together, following a ball in typical dog play fashion. This kind of activity is exactly what we'd expect to see when dogs are enjoying a game of ball chase together.

Figure 4. **Examples demonstrating increased response verbosity.** The model tends to generate overly detailed explanations for even simple multiple-choice questions. **Top:** When asked about horse direction change, the model provides an unnecessarily lengthy analysis of each option. **Bottom:** For a straightforward question about dogs playing, the response includes excessive behavioral analysis and option elimination. Red text highlights conclusive statements that could have been given directly (which are incorrect as well).



Q: Why does the woman in green pants skate behind everyone else?
Options: A. supervise the children B. singing C. pass her something D. waiting for the man E. nervous
Ans: A

VideoSAVi-Vicuna: The woman in green pants is likely skating behind everyone else to keep an eye on the children and ensure their safety as they enjoy their time on the snow. This is a responsible and caring action, as it allows the children to have fun while also ensuring their well-being.

VideoSAVi-Qwen: (A)

Q: What did the girl do after lifting the dog's ear in the middle of the video?
Options: A. starts to draw B. look at the baby C. push cheeks up D. smile at the camera E. wet the cloth
Ans: D

VideoSAVi-Vicuna: The girl in the video lifted the dog's ear and then petted the dog's face. After that, she continued to interact with the dog and pet it.

VideoSAVi-Qwen: (D)

Figure 5. **Examples demonstrating decreased instruction following in multiple-choice questions. Left:** While VideoSAVi-Qwen directly answers with the correct option "(A)", VideoSAVi-Vicuna provides a descriptive explanation without selecting from the given options. **Right:** Similarly, VideoSAVi-Vicuna elaborates on the action without explicitly choosing option "(D)" as requested. This behavior indicates compromised instruction-following capabilities in the Vicuna variant, likely due to conflicting optimization objectives between CLIP-adjusted DPO and SFT losses.

increase in entropy indicates that the model is less certain about its outputs, spreading probability mass over a wider range of possible responses. This behavior can manifest as longer, more detailed answers that attempt to cover multiple possible correct responses. Figure 4 shows an example of this where the responses are verbose and incorrect at the same time. These observations are in sync with the claims made in [41] regarding verbosity bias of DPO fine-tuning.

A.4.3. Decrease in Instruction Following

The SFT loss $\mathcal{L}_{\text{SFT}}(\theta)$ encourages the model to generate responses that are coherent and follow the instructions provided in the prompts. However, the conflicting optimization objectives introduced by the CLIP-adjusted DPO loss can impede this goal. Figure 5 shows that VideoSAVi-Vicuna fails to directly answer multiple-choice questions despite being explicitly prompted to select an option. This was the reason for reduced performance of Vicuna variants on multi-choice QA evaluation in Table 3.

Gradient Conflict. The total gradient is a weighted sum of the gradients from the CLIP-DPO loss and the SFT loss:

$$\nabla_\theta \mathcal{L}_{\text{Total}}(\theta) = \alpha \, \nabla_\theta \mathcal{L}_{\text{CLIP-DPO}}(\theta) + \gamma \, \nabla_\theta \mathcal{L}_{\text{SFT}}(\theta). \quad (12)$$

When $\nabla_\theta \mathcal{L}_{\text{CLIP-DPO}}$ and $\nabla_\theta \mathcal{L}_{\text{SFT}}$ point in opposing directions, the net gradient can be diminished, slowing down learning or causing the model to prioritize one objective over the other. This conflict can lead to decreased adherence to instructions, as the model balances between following the prompt and aligning with the visual content.

A.4.4. Increase in SFT Loss

Due to the gradient conflicts, the SFT loss may not decrease as expected during training. The model's capacity to learn from the SFT data is compromised, resulting in higher SFT loss values as shown in Figure 6. The SFT loss depends on the model's ability to predict the ground-truth tokens given the prompts. When the model is also being pushed to align with preferences that may conflict with the SFT data, its performance on the SFT objective can degrade.

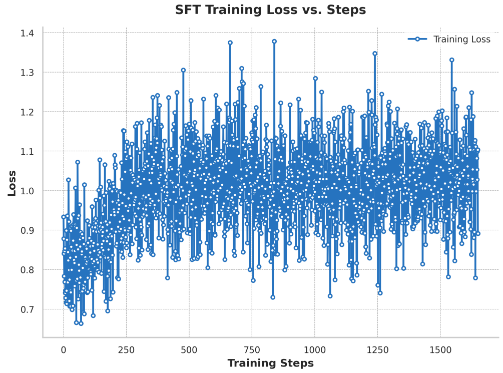


Figure 6. **Training dynamics of the SFT loss component.** The graph shows increasing SFT loss over training iterations when using CLIP-adjusted DPO, indicating the trade-off between visual alignment and language modeling performance. The higher SFT loss suggests compromised instruction-following capabilities as the model optimizes for noisy (model generated) preference data.

A.4.5. Policy Divergence and KL Regularization

The reward function $r_\theta(x, y)$ involves the log ratio between the policy model and the reference model. The KL divergence between the two models is given by:

$$D_{\text{KL}}\left(p_\theta(y|x) \,\|\, p_{\text{ref}}(y|x)\right) = \mathbb{E}_{y \sim p_\theta(y|x)}\left[\log \frac{p_\theta(y|x)}{p_{\text{ref}}(y|x)}\right]. \quad (13)$$

With the CLIP adjustment, the model may diverge more rapidly from the reference model, particularly for examples where $s_i = -1$. This divergence can lead to overfitting to the adjusted preferences, reducing the generalization capabilities of the model.

Method	LLM	Action	Direction	Speed	Event	Attribute	Average
		SigLIP-based Filtering					
VideoSAVi-Vicuna	7B	42.5	23.0	20.4	29.4	31.3	29.3
VideoSAVi-Qwen	.5B	47.5	41.8	43.0	35.7	46.0	42.8
VideoSAVi-Qwen	7B	83.4	43.3	48.0	47.1	50.6	54.5

Table 6. **Ablation study on TempCompass benchmark.** We examined the impact of swapping CLIP-based filtering with SigLIP-based filtering of the preference data. Results are reported across action, direction, speed, event, and attribute understanding tasks.

B. Ablation Study: SigLIP-based Filtering

We investigated the use of SigLIP [67] instead of CLIP for filtering DPO preference data. When filtering DPO preferences, both CLIP and SigLIP compute similarity scores between video frames and text pairs, but with crucial differences in their architectures and scoring mechanisms. We define the embedding functions:

- $f_{\text{CLIP}} : \mathcal{V} \to \mathbb{R}^{1536}$ is CLIP's ViT-L/14 visual encoder
- $g_{\text{CLIP}} : \mathcal{T} \to \mathbb{R}^{1536}$ is CLIP's text encoder
- $f_{\text{SigLIP}} : \mathcal{V} \to \mathbb{R}^{768}$ is SigLIP's ViT-B/16 vision transformer encoder
- $g_{\text{SigLIP}} : \mathcal{T} \to \mathbb{R}^{768}$ is SigLIP's text transformer encoder

where $\mathcal{V} = \mathbb{R}^{224\times224\times3}$ is the space of normalized image frames and $\mathcal{T}$ represents tokenized text sequences.

For a video frame sequence $\{v_i^{(t)}\}_{t=1}^{T}$ and response pair (y_i^+, y_i^-), CLIP computes raw cosine similarity:

$$c_i^{\text{CLIP}} = \frac{f_{\text{CLIP}}(v_i)^\top g_{\text{CLIP}}(y_i)}{\|f_{\text{CLIP}}(v_i)\|\|g_{\text{CLIP}}(y_i)\|} \in [-1, 1] \tag{14}$$

While SigLIP applies an additional sigmoid transformation:

$$c_i^{\text{SigLIP}} = \sigma\left(\frac{f_{\text{SigLIP}}(v_i)^\top g_{\text{SigLIP}}(y_i)}{\|f_{\text{SigLIP}}(v_i)\|\|g_{\text{SigLIP}}(y_i)\|}\right) \in (0, 1) \tag{15}$$

For a batch of frames $B = \{v_i^{(t)}\}_{t=1}^{|B|}$, we compute batch-wise similarities:

$$c_B^+ = \frac{1}{|B|}\sum_{v\in B} c_i(v, y_i^+), \quad c_B^- = \frac{1}{|B|}\sum_{v\in B} c_i(v, y_i^-) \tag{16}$$

The final filtering decision aggregates across all batches:

$$s_i = \begin{cases} +1, & \text{if } \frac{1}{N}\sum_{j=1}^{N} c_{B_j}^+ \geq \frac{1}{N}\sum_{j=1}^{N} c_{B_j}^-, \\ -1, & \text{otherwise} \end{cases} \tag{17}$$

where N is the total number of batches processed.

As shown in Table 6, this leads to notable drop in performance on the TempCompass benchmark [33] compared to Table 2, which used CLIP-based filtering: VideoSAVi-Vicuna (7B) dropped to 29.3% vs 47.7% average performance, and VideoSAVi-Qwen (.5B) saw a decrease to 42.8% vs 48.8% average performance. We hypothetize the performance degradation due to:

Similarity Range Compression: The sigmoid transformation in SigLIP compresses the similarity range:

$$\sigma : [-\infty, \infty] \to (0, 1) \quad \text{vs} \quad \cos : \mathbb{R}^d \times \mathbb{R}^d \to [-1, 1] \tag{18}$$

This makes it harder to distinguish between strongly aligned pairs, particularly impacting models with less capacity.

Non-linear Thresholding: The sigmoid's non-linearity affects the relative differences between similarity scores:

$$\frac{\partial\sigma(x)}{\partial x} = \sigma(x)(1 - \sigma(x)) \tag{19}$$

making the filtering decisions more sensitive to small variations in the raw similarity scores.

However, the larger 7B Qwen model's comparable performance between both filtering methods suggests sufficient capacity to produce consistent embeddings despite these transformations.

C. Filtering of Preference Data is Necessary

Figure 7, provides critical evidence for why CLIP filtering is essential in the VideoSAVi pipeline. The examples demonstrate instances where the model's preference selection, can diverge from visual ground truth. When evaluating response pairs, the model assigns preference scores based on semantic coherence and general knowledge, which can lead to visually inconsistent choices. In the macaron example, despite assigning a higher preference score (4.0) to the "four macarons" response, this preference contradicts the visual evidence. CLIP filtering serves as an independent vision-language verification mechanism, correctly identifying that the alternative "six macarons" response (scored 3.5) better aligns with the actual content. Similar behavior is observed in the architectural style assessment, where the model's preference for "flat roofs" (score 4.0) is corrected by CLIP filtering in favor of the visually accurate "traditional with snow-covered roofs" description. These cases highlight that model-based preference selection would generate noisy preference data for DPO and direct visual grounding through CLIP filtering is crucial for maintaining factual consistency. For evaluating individual responses, we employ a structured evaluation template shown in Figure 8, which ensures systematic scoring across multiple criteria including relevance, accuracy, temporal grounding, clarity, and groundedness.

Prompt: How many macarons are in the box?
Chosen: There are only **four macarons** in the box because each box contains a single color and has one macaron of that color.
Rejected: There are **six macarons** in the box.
Chosen score: 4.0
Rejected score: 3.5
CLIP chosen: "Rejected"

Prompt: What is the architectural style of the village in the video?
Chosen: The village in the video has some buildings featuring **flat roofs** and others with sloped ones.
Rejected: The architecture of the village in the video is **traditional** with **snow-covered roofs**, indicating a cold climate.
Chosen score: 4.0
Rejected score: 3.0
CLIP chosen: "Rejected"

Figure 7. **Examples demonstrating the necessity of CLIP filtering in preference selection.** The figure shows cases where model-assigned scores (Chosen/Rejected scores) diverge from actual video content, but CLIP filtering correctly identifies the more visually grounded response. **Left:** While the model prefers the "four macarons" response, CLIP filtering identifies that "six macarons" better matches the visual content. **Right:** Similarly, CLIP filtering corrects the model's preference for "flat roofs" in favor of the more accurate "traditional with snow-covered roofs" description. These examples highlight how CLIP filtering helps maintain vision-text alignment in the preference selection process.

You are an expert evaluator for video-based question-answering systems.

Video Caption:{caption}

Question: {question}

Model Predicted Answer: {answer}

Task:

Evaluate the model's predicted answer based on the following criteria:

1. Relevance: How well does the answer address the question?
2. Accuracy: Is the answer factually correct based on the video caption?
3. Temporal Grounding: Does the answer correctly reference the timing and sequence of events in the video?
4. Clarity: Is the answer clear, concise, and free from grammatical errors?
5. Groundedness: Does the answer rely solely on the information available in the video caption without introducing unsupported details or assumptions?

Important: The video caption is your only context for this video. Do not use any other information or context.

Output:

Provide only the overall score as a single number between 1 and 5, where 1 is the lowest and 5 is the highest. Do not include any additional text, explanations, or formatting. Ensure that the output follows the exact format below.

Examples:

- Score for this video : 4
- 3.0
- Score - 5

Figure 8. **Prompt template** used for evaluating generated responses.

Q: How did the man demonstrate to the baby to open his mouth before feeding him?
A: He opened his own mouth

LLaVA-NeXT-DPO: The man demonstrated to the baby to open his mouth by **using a milk bottle**.

LLaVA-HOUND-DPO: The man demonstrated to the baby how to open his mouth for feeding **by using his hands**. He gently held the baby's head and guided the baby's mouth open for feeding. This is a common practice during the early stages of a baby's feeding development.

i-SRT: The man in the video demonstrated to the baby how to open his mouth **for feeding by using his hands**. This is evident from the sequence of images where the man is seen guiding the baby's head and using his hands to open the baby's mouth. This method is often used to teach infants how to receive food from an adult, and it is a gentle and nurturing approach. The use of hands is also a common practice for feeding infants as it allows for a more personal and attentive interaction.

VideoSAVi: The man in the video demonstrated to the baby how to open his mouth by **opening his own mouth**, which is a natural way of modeling eating behavior that helps infants learn to receive food during feeding time.

Figure 9. **Qualitative example showing VideoSAVI's video understanding capabilities** on the Intent-QA [24] benchmark. VideoSAVi correctly identifies the natural demonstration method ("opening his own mouth"), while earlier models show varying interpretations from incorrect object identification (LLaVA-NeXT-DPO's [30] "milk bottle") to imprecise action descriptions (i-SRT [1] and LLaVA-HOUND-DPO's [70] "using hands"). This comparison demonstrates VideoSAVi's ability to capture fine-grained actions in videos.

Q: What notable similarities can be observed between the man and the woman's interactions with their cups throughout the video?
A: Both the man and the woman pick up their cups with their right hands, drink from them, and then place them back on the table.

LLaVA-NeXT-DPO: Both the man and the woman pick up their cups with their **left hands**, drink from them, and then place them back on the table.

LLaVA-HOUND-DPO: The notable similarities observed between the man and the woman's interactions with their cups throughout the video are that both individuals pick up their cups with their right hands, take a sip, and then **place them back on their laps gently**. This action is consistent and repetitive, indicating a shared routine or custom.

i-SRT: The notable similarities observed between the man and the woman's interactions with their cups throughout the video are that both individuals pick up their cups with their right hands, drink from them, and then **place them back on their laps gently**. This answer accurately captures the essence of their actions and the observed similarities between the man and the woman's behavior.

VideoSAVi: In the video, we can observe that both individuals display identical drinking patterns as they each **pick up their cups with their right hands, take a drink, and place them back on the table**.

Figure 10. **Qualitative examples showing VideoSAVi's video understanding capabilities** on the NExT-QA [59] benchmark. While LLaVA-NeXT-DPO [30] incorrectly identifies "left hands" for cup handling, both LLaVA-HOUND-DPO [70] and i-SRT [1] add unnecessary details to the description about "laps gently." In contrast, VideoSAVi provides precise and factual observations of the shared drinking patterns, accurately describing the sequence without adding unobserved details.