VidChain: Chain-of-Tasks with Metric-based Direct Preference Optimization for Dense Video Captioning

Anonymous submission

Abstract

Despite the advancements of Video Large Language Models (VideoLLMs) in various tasks, they struggle with fine-grained temporal understanding tasks, such as Dense Video Captioning (DVC). DVC is a complicated task of describing all events within a video while also temporally locating each event in a video, which integrates multiple fine-grained tasks, including video segmentation, video captioning, and temporal video grounding. Previous VideoLLMs attempt to solve DVC in a single step, failing to utilize their reasoning capability. Moreover, previous loss used for training VideoLLMs does not fully reflect evaluation metrics, therefore providing supervision not directly aligned to target tasks. To address such a problem, we propose a novel framework named VidChain comprised of Chain-of-Tasks (CoTasks) and Metric-based Direct Preference Optimization (M-DPO). CoTasks decompose a complex task into a sequence of sub-tasks, allowing VideoLLMs to leverage their reasoning capabilities more effectively. M-DPO aligns a VideoLLM with evaluation metrics, providing fine-grained supervision to each task that is well-aligned with metrics. Applied to two different VideoLLMs, VidChain consistently improves their fine-grained video understanding, thereby outperforming previous VideoLLMs on two different DVC benchmarks and also on the temporal video grounding task.

1 Introduction

With the rapid advancement of Large Language Models (LLMs), numerous studies (Liu et al. 2023; Dai et al. 2023; Liu et al. 2024) have incorporated LLMs into video understanding tasks, leading to the emergence of Video Large Language Models (VideoLLMs). These VideoLLMs (Li et al. 2023; Zhang, Li, and Bing 2023; Maaz et al. 2024) have demonstrated strong performance in various tasks such as video question answering and video captioning, showcasing their ability to understand and utilize visual information. Despite their success, recent studies (Ren et al. 2024; Huang et al. 2024; Qian et al. 2024) have revealed that VideoLLMs exhibit unsatisfactory performance when it comes to fine-grained temporal video understanding, which often require *multiple* video-related sub-tasks given a single untrimmed video.

We observe that VideoLLMs fall short of fine-grained temporal video understanding especially in Dense Video Captioning (DVC) due to two key reasons. First, the conventional practice in DVC of VideoLLMs employs one-step reasoning, which is known to be inferior to multi-step reasoning for complex tasks. In particular, existing VideoLLMs address DVC by predicting descriptions and timestamps of all events via a single-step generation. Second, the gap between training objectives (*e.g.*, next-token prediction) and evaluation metrics for DVC (*e.g.*, SODA) often leads to sub-optimal performance. The next-token prediction does not fully reflect the complex evaluation protocol which involves diverse metrics such as SODA, METEOR, and IoU.

To tackle these aforementioned issues, we introduce a novel framework, VidChain that enhances VideoLLMs' fine-grained temporal video understanding, comprised of Chain-of-Tasks (CoTasks), and Metric-based Direct Preference Optimization (M-DPO). First, we present CoTasks that decomposes the objective of the challenging task into a sequence of sub-task objectives. This simple decomposition enables the model to elicit its strong reasoning capability on DVC. Hence it eases the challenge of the complex task by solving only one sub-task at each step and further enhances its capability of fine-grained temporal video understanding. Second, to further align VideoLLM with the evaluation metrics of DVC, we present M-DPO which learns the *metric preference*, a preference based on the evaluation metric such as SODA, of each sub-task that composes DVC. Following the insight from DPO (Rafailov et al. 2023), which aligns LLM with human preferences, we adopt a similar approach yet we align VideoLLMs specifically with the metric preferences.

Interestingly, we observe that this simple adaptation of evaluation metrics provides two advantages: (1) it reduces the reliance on human annotators being cost-efficient. (2) metric evaluations expand beyond the standard binary decision dataset where the labels are *continuous e.g.*, 10.0, 8.5, 3.0, rather than *discrete e.g.*, win or lose. Moreover, we take account of the sequential sub-task prediction in CoTasks, where we supervise metric preferences on the *final* response of the model as well as on the *intermediate* sub-tasks that allow for more fine-grained supervision. Overall, our M-DPO is a novel method that reflects continuous characteristics of the metric-based evaluations into learning, and also provides intermediate task-specific supervision, further enhancing fine-grained video understanding of VideoLLMs. We evaluate our VidChain on two benchmarks-Activitynet

Captions and YouCook2 for the challenging DVC task, and Activitynet Captions for temporal video grounding (TVG).

In sum, our contributions are three-fold:

- We propose Chain-of-Tasks (CoTasks) that decomposes a complicated task into a sequence of sub-tasks, enabling the VideoLLM to elicit its strong reasoning capability to address the challenging task of DVC.
- We present Metric-based Direct Preference Optimization (M-DPO) that aligns VideoLLM with evaluation metrics for multiple fine-grained video understanding tasks, providing supervision targeted to each task.
- Our novel framework, VidChain comprising of CoTasks and M-DPO, is generally applicable to LLM-based models which consistently improves performances when applied to baseline models.

2 Related works

Video Large Language Models. Recently, multiple works (Liu et al. 2023; Dai et al. 2023; Liu et al. 2024; Chen et al. 2024; Ye et al. 2024) incorporating Large Language Models (LLMs) for vision-language tasks have been proposed. Following those models' successes, several Video Large Language Models (VideoLLMs) have been proposed (Li et al. 2023; Zhang, Li, and Bing 2023; Maaz et al. 2024; Zhu et al. 2024; Lin et al. 2023; Li et al. 2024). Despite the remarkable performance of VideoLLMs in tasks requiring a holistic understanding of a video (*e.g.*, video-level question-answering or captioning), they often fall short in fine-grained video understanding. For instance, they often suffer in temporal grounding tasks (Krishna et al. 2017) or dense video captioning tasks (Krishna et al. 2017; Zhou, Xu, and Corso 2018), where diverse fine-grained video understanding capabilities are required. Thus, multiple works (Ren et al. 2024; Huang et al. 2024; Qian et al. 2024) have tried incorporating fine-grained information into VideoLLMs to address the problem. In this study, we propose decomposing a complicated task of DVC into simpler sub-tasks and providing supervision aligned with the desired capability, thereby enhancing VideoLLMs' capability in fine-grained understanding.

Direct Preference Optimization. To align LLM outputs with human preferences, reinforcement learning from human feedback (RLHF) (Christiano et al. 2017; Ouyang et al. 2022) has been proposed, which maximizes the likelihood gap between the preferred and unpreferred generation results. Direct preference optimization (DPO) (Rafailov et al. 2023) is derived to improve the inefficiency of RLHF, lifting the need for RL-based optimization and dedicated modules (*i.e.*, reward model), which is applied to various tasks (Song et al. 2024; Xu et al. 2024; Yuan et al. 2024) to inject human preferences to a model. In recent, some works have applied preference alignment in multimodal language models (MLLMs) to alleviate the hallucination issue (Sun et al. 2023; Yu et al. 2024; Ahn et al. 2024; Gunjal, Yin, and Bas 2024). However, these approaches rely on expensive models like GPT-4v (Ahn et al. 2024) or human annotators (Yu et al. 2024) to annotate preference data. In this work, we adopt the idea of Step-DPO (Lai et al. 2024) to align VideoLLM on every sub-task with the desired capability in fine-grained video understanding by defining the preferred and unpreferred responses using the metric as a criterion. Such an approach eliminates the need for extensive human labor or computation. Also, unlike conventional DPO datasets where only a binary preference exists, a continuous preference exists in our dataset due to the continuous nature of metrics that we build upon. Therefore, we further propose a tailored training scheme reflecting the continuous nature of preferences.

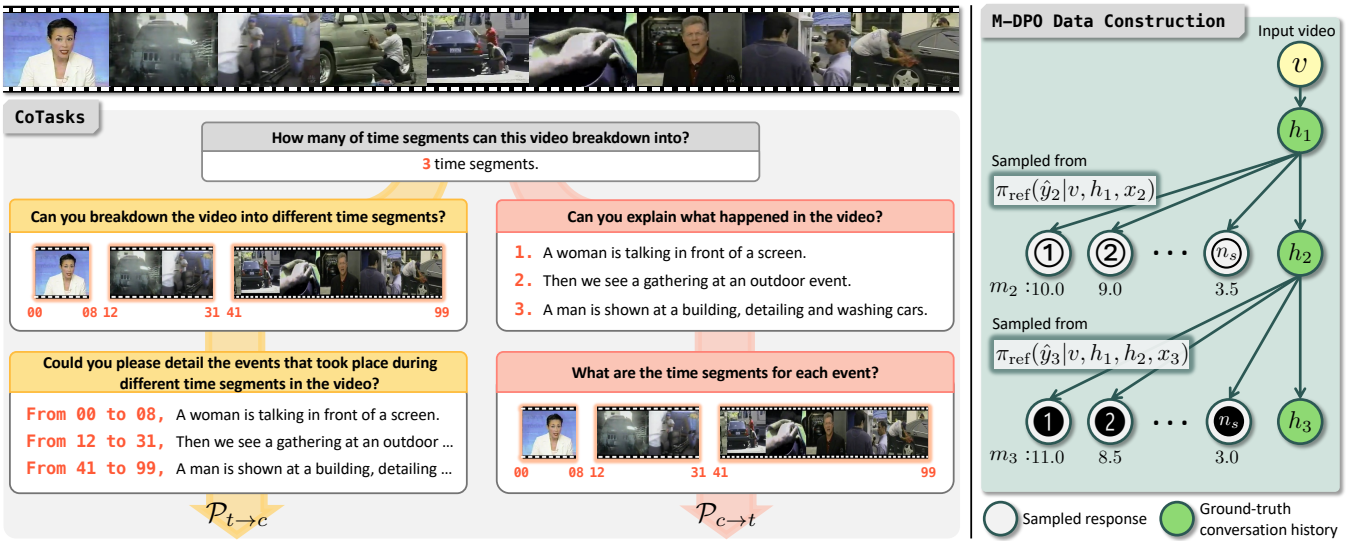
3 Method

In this section, we first provide a brief overview of Dense Video Captioning (DVC) and Direct Preference Optimization (DPO) in Sec. 3.1. Then, we propose a Chain-of-Tasks (CoTasks) approach which eases the challenge of DVC by decomposing the task into a sequence of sub-tasks (Sec. 3.2). We then present a Metric-based DPO (M-DPO), which further aligns VideoLLMs with evaluation metrics for sub-tasks (*e.g.,* METEOR, IoU, SODA) to provide more fine-grained supervision (Sec. 3.3). Comprised of CoTasks and M-DPO, we propose a novel framework named VidChain that enhances VideoLLMs' fine-grained temporal video understanding capability.

3.1 Preliminaries

Dense Video Captioning. Dense Video Captioning (DVC) is a challenging task that requires the model to not only describe all events within the long untrimmed video but also temporally localize each event in time. Given an untrimmed video v, the goal of DVC is to maximize the probability $p(c, t, n|v)$, where n denotes the number of events in the video, c denotes a set of event captions, and t denotes a set of event timestamps represented with start and end time boundaries for each. The key challenge is that the model must predict the three components (c, t, n) for all events given an untrimmed video v, which requires a comprehensive fine-grained understanding regarding multiple video-related tasks: video segmentation, video captioning, and temporal video grounding. The video segmentation task aims to predict the number of event sequences the video breaks down into. Video captioning focuses on describing the events in the video. The temporal video grounding task aims to identify the timestamps of an event given its event description. Typically, VideoLLMs address DVC by predicting (c, t, n) in a single step, which imposes more challenge on the task.

Direct Preference Optimization. Direct Preference Optimization (DPO) (Rafailov et al. 2023) aligns Large Language Models' output with human preferences. Often, the alignment involves further finetuning a supervised finetuned model. For optimization, DPO adopts a pairwise preference dataset, where each sample comprises a pair of preferred and dispreferred responses. To construct the pairwise preference dataset, several responses $\hat{y}$ are first sampled from the reference model π_{ref} given a prompt x. Then, those responses are annotated according to the human preferences by comparing sampled responses in a pair-wise manner where $\hat{y}^w$ and $\hat{y}^l$



(a) (b) (c)

Figure 1: **Illustration of our CoTasks approach (left) and data construction process for M-DPO (right).** The left figure depicts the CoTasks approach of VidChain, which decomposes DVC into a sequence of sub-tasks in two different reasoning paths. After predicting the number of events, timestamp prediction and caption generation are done in path $\mathcal{P}_{t\to c}$ as shown in (a), while the order of two tasks is interchanged in path $\mathcal{P}_{c\to t}$ as in (b). The right figure (c) represents the data construction of M-DPO, where we sample n_s response of $\hat{y}_3$ (filled black circles) as well as the intermediate sub-task response $\hat{y}_2$ (hollow black circles) of CoTask for the given video v. The m_2 and m_3 denote the task-specific evaluation metric values, *e.g.*, SODA$_c$, METEOR, IoU, of each sampled response.

denote the preferred and dispreferred response respectively, *i.e.*, $\hat{y}^w \succ \hat{y}^l|x$. With the constructed pairwise preference dataset $\mathcal{D}_{\text{DPO}}$, the objective of DPO, $\mathcal{L}_{\text{DPO}}$, is formally defined as follows:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta;\pi_{\text{ref}}) = \\ -\mathbb{E}_{(x,\hat{y}^w,\hat{y}^l)\sim\mathcal{D}_{\text{DPO}}}\left[\log\sigma\Big(\beta\cdot r(\hat{y}^w;x)-\beta\cdot r(\hat{y}^l;x)\Big)\right], \quad (1)$$

where $r(y;x)=\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$. Note that $\sigma(\cdot)$ is the sigmoid function, π_θ is the model to be optimized, which is initialized to π_{ref}, and β is a hyperparameter that controls the distribution disparity of π_θ from the reference model π_{ref}. Overall, the model is trained to increase the likelihood of the preferred responses relative to that of dispreferred responses.

3.2 Chain-of-Tasks (CoTasks).

To address the lack of fine-grained temporal understanding of VideoLLMs, especially in DVC that encompasses multiple video-related tasks, we propose a novel approach named Chain-of-Tasks. Most prior works (Ren et al. 2024; Qian et al. 2024; Huang et al. 2024; Yang et al. 2023) address DVC by directly predicting (c,t,n) given v within a single step. Yet, this approach imposes more challenges on the task for the VideoLLM, as it obstructs the model from leveraging its strong reasoning capability. Hence, in CoTasks, we first decompose the objective of DVC into a series of sequential sub-task objectives. Then we prompt each task corresponding to each objective to the VideoLLM in the form of a multi-turn QA conversation. Such an approach eases the challenge of DVC by solving only one sub-task at each turn and further enhances a VideoLLM's capability in fine-grained temporal video understanding.

Objective Decomposition. The objective of DVC can be decomposed in two different reasoning paths, $\mathcal{P}_{t\to c}$ and $\mathcal{P}_{c\to t}$:

$$p(c,t,n|v) = p(c|v,n,t)p(t|v,n)p(n|v) \quad (\mathcal{P}_{t\to c}) \quad (2)$$
$$= p(t|v,n,c)p(c|v,n)p(n|v) \quad (\mathcal{P}_{c\to t}). \quad (3)$$

In the case of $\mathcal{P}_{t\to c}$ in Eq. (2), the prediction of (c,t,n) given a video v breaks down into three sequential tasks. First, $p(n|v)$ represents the task of predicting the number of events in the video, while the following $p(t|v,n)$ represents the task of the timestamp prediction given the total number of events n, and the final $p(c|v,n,t)$ indicates caption generation for each event given the video with its t and n. The other path $\mathcal{P}_{c\to t}$ in Eq. (3) is also similarly defined, except that the order of the caption generation and the timestamp prediction tasks are interchanged. Based on this decomposition, we cast the task of DVC as a multi-turn prediction, where the model sequentially solves different tasks at each turn to tackle the challenging task. An example of our multi-turn approach, namely CoTasks, is illustrated in Fig. 1.(a) and 1.(b), each corresponding to $\mathcal{P}_{t\to c}$, and $\mathcal{P}_{c\to t}$.

Training data construction for CoTasks. In this section, we elaborate on the construction process of $\mathcal{D}_{\text{CT}}$, a multi-turn conversation dataset used for training VideoLLMs to reason in a CoTasks manner. We build CoTasks samples using the original DVC dataset (*e.g.*, ActivityNet or YouCook2), by converting the original single-turn conversation samples into multi-turn CoTasks samples of both

$\mathcal{P}_{t\to c}$ and $\mathcal{P}_{c\to t}$ types. We construct 10K and 1K samples for AcitivtyNet and YouCook respectively for each path using the pre-defined templates, where the templates are provided in the supplementary material. Note we refer to each of the two types of dataset as $\mathcal{D}_{t\to c}$ and $\mathcal{D}_{c\to t}$, respectively. Combining our obtained $\mathcal{D}_{t\to c}$ and $\mathcal{D}_{c\to t}$ with the dataset used for training VTimeLLM (Huang et al. 2024) to best follow their training protocols while adopting the full benchmark dataset, $\mathcal{D}_{\text{CT}}$ with a size of 50K for ActivityNet and 6K for YouCook2 is constructed. More details of $\mathcal{D}_{\text{CT}}$ are in the supplement. Then we use $\mathcal{D}_{\text{CT}}$ to finetune VideoLLMs, resulting in VideoLLMs that are better at multiple fine-grained video understanding tasks, including DVC and its sub-tasks. By utilizing LoRA for parameter-efficient fine-tuning, the most time-consuming experiment was done in just 6 hours using 8 RTX A6000 GPUs.

Inference pipeline of CoTasks. Since $\mathcal{D}_{\text{CT}}$ includes both samples of $\mathcal{D}_{t\to c}$ and $\mathcal{D}_{c\to t}$, the VideoLLM trained with $\mathcal{D}_{\text{CT}}$ can take either reasoning path during inference to address DVC. To encourage the model to take a certain path, we prompt the model with path-specific prompts. In other words, for the path of $\mathcal{P}_{c\to t}$, we prompt "Can you explain what happened in the video?" to encourage the generation of event captions first, after addressing the common task for both paths, *i.e.*, the number of event predictions. In our experiments, we provide results in both inference paths.

3.3 Metric-based Direct Preference Optimization

Although CoTasks enhances VideoLLMs in fine-grained video understanding tasks, the next-token prediction objective does not fully reflect the complex evaluation protocol which involves diverse metrics such as SODA, METEOR, and IoU. Therefore, we propose a novel optimization method named Metric-based Direct Preference Optimization (M-DPO). Inspired by DPO, which aligns a model with human preferences, M-DPO aligns the VideoLLM with metric preferences using pairs of a preferred and a dispreferred response, where the evaluation metric for tasks within CoTasks is adopted as criteria to determine preferred and dispreferred responses. This approach enables more fine-grained metric preference alignment of the VideoLLM as it not only supervises the *final* response but also across the *intermediate* responses within CoTasks. In the following sections, we first describe the process of constructing a dataset used for M-DPO training, where a sample comprises pairs of preferred and dispreferred responses using metrics as criteria. Then, we introduce the overall training objective of M-DPO. Finally, we present a preference gap-aware M-DPO, which is an extension of M-DPO equipped with a tailored training scheme reflecting the continuous nature of metrics.

Training data construction for M-DPO. In this section, we elaborate on the process of constructing $\mathcal{D}_{\text{M-DPO}}$, a preference dataset used for further aligning a VideoLLM with metric preferences, where each sample includes a pair of preferred and dispreferred responses for each specific task in the CoTask approach. To obtain a preference pair, n_s number of responses are first sampled for each intermediate k-th task given a video v, prompt x_k for k-th task, and the conversation history $h_{<k}$ that consists of prompts $x_{<k}$ and responses $y_{<k}$ of previous k tasks. Starting from $k=2$, a single sampled response $\hat{y}_k$ is represented as:

$$\hat{y}_k \sim \pi_{\text{ref}}(\hat{y}_k|v, h_{<k}, x_k) \tag{4}$$

where the reference model π_{ref} is a VideoLLM trained with $\mathcal{D}_{\text{CT}}$ for CoTasks, and $h_{<2}$ is the ground-truth conversation history consistsing of the prompt x_1 and ground-truth response y_1 corresponding to $p(n|v)$. For instance, in $\mathcal{P}_{t\to c}$ path of CoTasks, $\hat{y}_3$ is a response sampled from $\pi_{\text{ref}}(\hat{y}_3|v, h_{<3}, x_3)$, which models $p(c|v, n, t)$. Similarly, $\hat{y}_2$ is a response sampled from $\pi_{\text{ref}}(\hat{y}_2|v, h_{<2}, x_2)$, modeling $p(t|v, n)$. With the n_s sampled responses for each task, $\binom{n_2}{2}$ pairs of responses are obtained. Then, for each pair, a response $\hat{y}_k$ with higher evaluation metric $m_k = \mathcal{M}_k(\hat{y}_k, y_k)$ is set as a preferred response $\hat{y}_k^w$, and the other response is set as a dispreferred response $\hat{y}_k^l$, where $\mathcal{M}_k$ denotes a metric corresponds to k-th task (*i.e.*, METEOR, IoU, SODA$_c$). In other words, $\hat{y}_k^w \succ \hat{y}_k^l | v, h_{<k}, x_k$, given $m_k^w > m_k^l$.

Overall, a sample $d_{\text{M-DPO}} \sim \mathcal{D}_{\text{M-DPO}}$ is defined as a tuple consists of pair of responses $\hat{y}_k^w, \hat{y}_k^l$, metrics m_k^w, m_k^l of each response, ground-truth conversation history $h_{<k}$ up to $(k-1)$-th task, prompt x_k for the k-th task, and a corresponding video v. The illustration of our M-DPO training dataset construction process is in Fig 1. (c).

Training objective of M-DPO. With the M-DPO dataset $\mathcal{D}_{\text{M-DPO}}$ obtained as described above, the VideoLLM is further trained to align with the metric preferences. Formally, the M-DPO loss regarding a single data $d_{\text{M-DPO}}$ from $\mathcal{D}_{\text{M-DPO}}$ is defined as:

$$\begin{aligned}&\mathcal{L}_s(y_k^w, y_k^l; v, h_{<k}, x_k) = \\ &\left[\log\sigma\left(\beta r(\hat{y}_k^w; v, h_{<k}, x_k) - \beta r(\hat{y}_k^l; v, h_{<k}, x_k)\right)\right],\end{aligned} \tag{5}$$

where σ denotes the sigmoid function, $r(\hat{y}_k; v, h_{<k}, x_k) = \log\frac{\pi_\theta(\hat{y}_k|v,h_{<k},x_k)}{\pi_{\text{ref}}(\hat{y}_k|v,h_{<k},x_k)}$ denotes a likelihood ratio, π_θ denotes the target model to be optimized, and β is a hyperparameter controlling the distribution disparity of π_θ from the reference model π_{ref}. Thus, by minimizing the given loss, it encourages the model to learn the metric-based preference on the k-th task by enlarging the gap of the likelihood ratio between preferred and dispreferred responses in terms of the target metric. Then, the basic version of M-DPO training objective is defined as below, of which the M-DPO loss is averaged over every sample in $\mathcal{D}_{\text{M-DPO}}$:

$$\begin{aligned}&\mathcal{L}_{\text{M-DPO}^-}(\pi_\theta; \pi_{\text{ref}}) = \\ &-\mathbb{E}_{d_{\text{M-DPO}}\sim\mathcal{D}_{\text{M-DPO}}}\left[\mathcal{L}_s\left(\hat{y}_k^w, \hat{y}_k^l; v, h_{<k}, x_k\right)\right],\end{aligned} \tag{6}$$

where π_θ is a model to be optimized, which is built by adding LoRA modules after the initialization with π_{ref}. Note that LoRA modules in π_θ are only trainable parameters, and π_{ref} is left unchanged.

Preference gap-aware M-DPO. Training data for conventional DPO only includes *binary* preferences $\hat{y}^w$ and $\hat{y}^l$, which only indicates whether a response is preferred or not. On the contrary, data in $\mathcal{D}_{\text{M-DPO}}$ also comprises *continuous*

	LM size	ActivityNet SODA$_c$	ActivityNet METEOR	ActivityNet CIDEr	YouCook2 SODA$_c$	YouCook2 METEOR	YouCook2 CIDEr
VideoChat (Li et al. 2023)	7B	0.9	0.9	2.2	-	-	-
VideoLLaMA (Zhang, Li, and Bing 2023)	7B	1.9	1.9	5.8	-	-	-
VideoChatGPT (Maaz et al. 2024)	7B	1.9	2.1	5.8	-	-	-
TimeChat (Ren et al. 2024)	13B	-	-	-	3.4	-	11.0
VTimeLLM (Huang et al. 2024)	13B	5.9	6.7	27.2	-	-	-
VTimeLLM† (Huang et al. 2024) (Baseline)	7B	5.8	6.8	27.6	3.4	3.5	10.7
VTimeLLM + VidChain-$\mathcal{P}_{t\to c}$ **(Ours)**	7B	6.5	**7.4**	28.2	**4.6**	**4.9**	**17.6**
VTimeLLM + VidChain-$\mathcal{P}_{c\to t}$ **(Ours)**	7B	**6.6**	7.2	**29.8**	4.3	4.5	16.3
VideoLLaMA2† (Cheng et al. 2024) (Baseline)	7B	7.2	7.7	32.9	3.3	3.5	12.3
VideoLLaMA2 + VidChain-$\mathcal{P}_{t\to c}$ **(Ours)**	7B	8.2	8.7	43.1	4.6	5.5	22.3
VideoLLaMA2 + VidChain-$\mathcal{P}_{c\to t}$ **(Ours)**	7B	**8.8**	**8.8**	**43.9**	**4.8**	**5.6**	**23.8**

Table 1: **Comparison of VideoLLMs on DVC.** Baseline+VidChain-$\mathcal{P}_{t\to c}$ and Baseline+VidChain-$\mathcal{P}_{c\to t}$ are identical models trained with $\mathcal{D}_{\text{CT}}$ which adopt two different reasoning path prompts for inference, $\mathcal{P}_{t\to c}$ and $\mathcal{P}_{c\to t}$ respectively. See Sec. 3.2 for more detail. † denotes reproduced results.

	LM size	R@0.3	R@0.5	R@0.7	mIoU
VideoChat	7B	8.8	3.7	1.5	7.2
VideoLLaMA	7B	6.9	2.1	0.8	6.5
VideoChatGPT	7B	26.4	13.6	6.1	18.9
TimeChat	13B	-	-	-	-
VTimeLLM	13B	44.8	29.5	14.2	31.4
VTimeLLM (Baseline)	7B	44.0	27.8	14.3	30.4
VTimeLLM + VidChain **(Ours)**	7B	**63.3**	**47.0**	**29.5**	**45.5**
VideoLLaMA2 (Baseline)	7B	49.4	26.8	15.0	33.9
VideoLLaMA2 + VidChain **(Ours)**	7B	**63.3**	**44.8**	**25.2**	**44.1**

Table 2: **Comparison of VideoLLMs on TVG.** We simply adopt the task-specific prompt for TVG instead of two different inference prompts (*i.e.*, $P_{t\to c}$, and $P_{c\to t}$) specifically defined for DVC, since they are not applicable to TVG.

preferences m_k^w and m_k^l which not only indicates whether a response is preferred or not but also reveals *how much* a response is preferred since it is built on continuous metrics. We observe that when optimizing with such continuous preferences, taking the gap of preferences between $\hat{y}_k^w$ and $\hat{y}_k^l$ into account further facilitates the proper training. To this end, we propose $\mathcal{L}_{\text{M-DPO}}$, an advanced version of $\mathcal{L}_{\text{M-DPO}^-}$ by modifying Eq. (6) as:

$$\mathcal{L}_{\text{M-DPO}}(\pi_\theta;\pi_{\text{ref}}) = -\mathbb{E}_{d_{\text{M-DPO}}\sim\mathcal{D}_{\text{M-DPO}}} \left[\mathbb{1}\left(m_k^w - m_k^l > \gamma\right)\cdot\mathcal{L}_s\left(\hat{y}_k^w,\hat{y}_k^l;v,h_{<k},x_k\right)\right], \tag{7}$$

where $\mathbb{1}(\cdot)$ denotes an indicator function. Concretely, we only calculate losses on preference pairs where the gap of the evaluation metrics between the preferred and dispreferred response is above a certain threshold γ. Such an approach alleviates difficulties in optimizing pairs with subtle differences in metrics, thereby facilitating the overall optimization process. In the following sections, the term 'M-DPO' refers to $\mathcal{L}_{\text{M-DPO}}$ in Eq. (7) instead of $\mathcal{L}_{\text{M-DPO}^-}$ in Eq. (6) unless specified. Training with M-DPO is also efficiently done in 9 hours with 8 RTX A6000 GPUs in the most time-consuming experiment. Overall, we propose a novel framework named VidChain comprised of CoTasks and M-DPO which effectively enhances the fine-grained temporal video understanding of VideoLLMs.

	ActivityNet SODA$_c$	ActivityNet METEOR	YouCook2 SODA$_c$	YouCook2 METEOR
VTimeLLM				
Baseline	5.8	6.8	3.4	3.5
+ CoTasks-$\mathcal{P}_{t\to c}$	**6.5**	7.1	4.1	4.4
+ VidChain-$\mathcal{P}_{t\to c}$	**6.5**	**7.4**	**4.6**	**4.9**
+ CoTasks-$\mathcal{P}_{c\to t}$	6.5	**7.3**	3.8	4.3
+ VidChain-$\mathcal{P}_{c\to t}$	**6.6**	7.2	**4.3**	**4.5**
VideoLLaMA2				
Baseline	7.2	7.7	3.3	3.5
+ CoTasks-$\mathcal{P}_{t\to c}$	7.5	8.3	4.2	5.1
+ VidChain-$\mathcal{P}_{t\to c}$	**8.2**	**8.7**	**4.6**	**5.5**
+ CoTasks-$\mathcal{P}_{c\to t}$	7.7	8.5	4.5	5.5
+ VidChain-$\mathcal{P}_{c\to t}$	**8.8**	**8.8**	**4.8**	**5.6**

Table 3: **Ablation study on components of VidChain.** VidChain denotes CoTasks + M-DPO.

4 Experiments

4.1 Benchmarks

Dense Video Captioning. We experiment on two different dense video captioning benchmarks, ActivityNet Captions (Krishna et al. 2017) and YouCook2 (Zhou, Xu, and Corso 2018). ActivityNet Captions dataset consists of 20k videos annotated with temporally localized descriptions. YouCook2 dataset is composed of 2,000 videos from 89 recipes. Each video has temporal bounds and their corresponding context sentences. As evaluation metrics, SODA$_c$ (Fujita et al. 2020), METEOR (Banerjee and Lavie 2005), and CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015) following previous works (Huang et al. 2024; Qian et al. 2024).

Temporal Video Grounding. Temporal video grounding is the task of localizing multiple events in the video given captions of each event. For temporal video grounding, we use ActivityNet Captions dataset (Krishna et al. 2017) to validate the effectiveness of our method. As evaluation metrics,

	Training Data ($\mathcal{D}_{\text{CT}}$)		Dense Video Captioning	
	$\mathcal{D}_{t\to c}$	$\mathcal{D}_{c\to t}$	SODA$_c$	METEOR
Baseline	✘	✘	7.2	7.7
CoTasks-$\mathcal{P}_{t\to c}$	✔	✘	7.4	7.6
	✔	✔	**7.5**	**8.3**
CoTasks-$\mathcal{P}_{c\to t}$	✘	✔	7.6	8.1
	✔	✔	**7.7**	**8.5**

Table 4: **Ablation study on data composition of $\mathcal{D}_{\text{CT}}$.**

R@$\{0.3, 0.5, 0.7\}$ and mIoU are applied following the previous works (Huang et al. 2024; Qian et al. 2024). For implementation details and further details about metrics, refer to the supplementary material.

4.2 Main results

We evaluate VidChain on the challenging Dense Video Captioning (DVC) and temporal video grounding (TVG) to verify the effectiveness of our approach in enhancing the fine-grained video understanding. Note we report performances for both CoTasks paths, $\mathcal{P}_{c\to t}$ and $\mathcal{P}_{t\to c}$ for the DVC task.

Results. Table 1 demonstrates the effectiveness of the proposed VidChain by applying it on two state-of-the-art VideoLLMs, VTimeLLM and VideoLLaMA2. VidChain improves both VideoLLMs on two DVC benchmarks, ActivityNet and YouCook, thereby outperforming every VideoLLM. In detail, VideoLLaMA2+VidChain-$\mathcal{P}_{c\to t}$ shows a +22.2% gain in SODA$_c$ increasing from 7.2 to 8.8, 14.3% gain in METEOR and as much as 33.4% in CIDEr on ActivityNet. In YouCook2, the model shows an 45.5%, 60%, 93.5% increase for SODA$_c$, METEOR, and CIDEr, respectively.

Similar to the case of VideoLLaMA2, VidChain also shows consistent performance gains with VTimeLLM. For instance, VidChain boosts the performance of VTimeLLM by up to 1.2, 1.4, and 6.9 points in SODA$_c$, METEOR, and CIDEr respectively on the YouCook benchmark for DVC, while it also outperforms the baseline in the ActivityNet in every metrics. Notably, VidChain applied to VTimeLLM with 7B LLM outperforms the baseline VTimeLLM with 13B LLM on every task by a large margin.

Moreover, Tab. 2 demonstrates the effectiveness of VidChain on TVG, where we show a prominent increase in performance when applied to both VideoLLMs. In particular, VTimeLLM+VidChain shows a 19.3, 19.2, 15.2, and 15.1 increase in Recall@0.3, Recall@0.5, Recall@0.7, and mIoU. The enhanced performance on TVG underlines the effectiveness of VidChain in enhancing the capability of a VideoLLM fine-grained video understanding, thereby also improving performance on a sub-task for DVC.

4.3 Quantitative Analysis

In the following experiments and analysis, we report results on ActivityNet for DVC using our best-performing model VideoLLaMA2+VidChain unless specified.

Effectiveness of CoTasks. In Tab. 3, we analyze the effectiveness of CoTasks. Results show that CoTasks yields consistent performance improvement on both VTimeLLM and VideoLLaMA2 regardless of the inference path ($\mathcal{P}_{t\to c}$ or $\mathcal{P}_{c\to t}$), compared to the baselines. In particular, when applied to VTimeLLM, CoTasks-$\mathcal{P}_{t\to c}$ shows 12.1% gain in SODA$_c$ for ActivityNet, and 20.6% gain in YouCook2. Moreover, we also observe consistent gains in TVG tasks by applying CoTasks, where the results are in the supplementary material. This result verifies the effectiveness of CoTasks in enhancing the reasoning capability of a VideoLLM.

	DVC		TVG	
	SODA$_c$	METEOR	R@0.3	mIoU
Baseline	7.7	8.5	60.2	41.9
$\mathcal{L}_{\text{DPO}}$	8.3	8.6	61.6	42.8
$\mathcal{L}_{\text{M-DPO}^-}$	8.6	**8.8**	62.4	43.4
$\mathcal{L}_{\text{M-DPO}}$ (Ours)	**8.8**	**8.8**	**63.3**	**44.1**

Table 5: **Analysis on DPO objectives.** Note the baseline refers to VideoLLaMA2+CoTasks-$\mathcal{P}_{c\to t}$.

Effectiveness of M-DPO. In Tab. 3, results of +VidChain are also reported, which denotes that M-DPO is also applied on top of CoTasks. Our results show that M-DPO generally improves performance on both VideoLLMs, VTimeLLM, and VideoLLaMA2. For instance, further training VideoLLaMA2+CoTasks-$\mathcal{P}_{c\to t}$ with M-DPO improves SODA$_c$, and METEOR by 1.1, and 0.3 points, as shown by the result of VideoLLaMA2+VidChain-$\mathcal{P}_{c\to t}$. Similarly, applying M-DPO consistently boosts performance on every VideoLLMs and benchmarks, showing its effectiveness.

Ablation study on data composition of $\mathcal{D}_{\text{CT}}$. We conduct an ablation study on the effect of the inclusion of data with two paths for DVC, namely $\mathcal{D}_{t\to c}$ and $\mathcal{D}_{c\to t}$ in CoTasks training data $\mathcal{D}_{\text{CT}}$. The results are in Tab. 4, where including data with both paths is shown to perform better than only utilizing a single type of path. We conjecture that two different paths are complementary to each other, therefore composing $\mathcal{D}_{\text{CT}}$ with data in both paths facilitates a VideoLLM's fine-grained video understanding capacity by letting a VideoLLM learn to solve the same objective in different ways.

Analysis on DPO objectives. In Tab. 5, an analysis of different DPO objectives is presented. Every experiment is done with $\mathcal{D}_{\text{M-DPO}}$ dataset constructed by ours, while the optimization objective applied differs. $\mathcal{L}_{\text{DPO}}$ (row 2) denotes that DPO is only applied to the final task instead of intermediate tasks, and preference-gap aware DPO is not applied. Overall, it generally improves DVC performance, showing the effectiveness of the DPO approach in the DVC task. Still, additionally optimizing intermediate tasks with $\mathcal{L}_{\text{M-DPO}^-}$ (row 3) enables additional gains of 0.3, and 0.2 over $\mathcal{L}_{\text{DPO}}$ (row 2) in SODA$_c$, and METEOR. Finally, applying the preference gap-aware M-DPO ($\mathcal{L}_{\text{M-DPO}}$) results in the best performance (row 4). The ablation results show the effectiveness of M-DPO components.

Analysis on DPO margin. In Fig. 3, the margin of likelihood ratio between preferred and dispreferred responses under different DPO objectives is plotted by epoch. A larger

Single-turn

Q: "Could you outline the incidents that occurred at various timestamps in the video?"

A: A woman is seen sitting on a couch with a cat in her lap, from **00 to 17**. She clips the cat's nails and then pets him, from **17 to 33**. She then clips the cat's other paw and pets him, from **33 to 50**. She clips the cat's other paw and pets him, from **33 to 50**. She clips the cat's other paw and pets him, from **67 to 83**. She clips the cat's other paw and pets him, from **83 to 99**.

→ ***Wrong number of events & repetitive captions***

CoTasks	VidChain (CoTasks + M-DPO)
Q1: "How many of time segments can this video breakdown into?" **A1:** 3 time segments.	**Q1:** "How many of time segments can this video breakdown into?" **A1:** 3 time segments.
Q2: "Can you explain what happened in the video?" **A2: 1.** A woman is sitting on a couch with a cat in her lap. **2.** She holds up a pair of clippers and a pair of scissors. **3.** She then proceeds to cut the cat's claws.	**Q2:** "Can you explain what happened in the video?" **A2: 1.** A woman is seen sitting on a couch holding a pair of clippers and speaking to the camera. → ***Closest to ground truth*** **2.** The woman then cuts the nails of a cat in her lap while the cat moves it's feet. **3.** The woman continues cutting the nails and is shown speaking to the camera.
Q3: "What are the time segments for each event?" **A3:** First event, from **00 to 11**. Second event, from **12 to 55**. Third event, from **56 to 99**.	**Q3:** "What are the time segments for each event?" **A3:** First event, from **00 to 25**. Second event, from **25 to 74**. Third event, from **73 to 99**.

Ground Truth

[00, 25]: A woman is seen holding on a cat on a coach and holding up a pair of nail clippers.
[25, 72]: She begins cutting the cat's claws while the cat squirms around a bit.
[72, 93]: She continues cutting the claws while speaking to the camera.

Figure 2: **Qualitative example of Dense Video Captioning.** Predictions of baseline VideoLLM (Single-turn), VideoLLM+CoTasks, and VidChain (CoTasks + M-DPO) are illustrated. Red and green highlights denote erroneous and accurate predictions, respectively. Visualization is done on ActivityNet validation set with VTimeLLM in $\mathcal{P}_{c\to t}$ path.

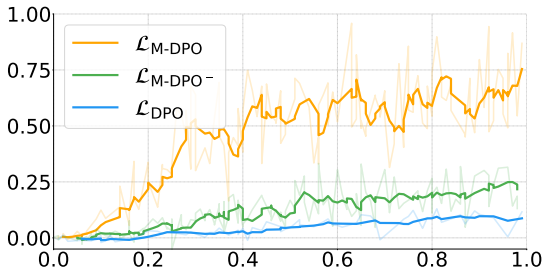


Figure 3: **Margin of the likelihood ratio between preferred and dispreferred responses** with $\mathcal{L}_{\text{DPO}}$, $\mathcal{L}_{\text{M-DPO}^-}$, and $\mathcal{L}_{\text{M-DPO}}$. x-axis stands for training epochs.

margin implies that preferred and dispreferred responses are clearly distinguished by a model. As illustrated, $\mathcal{L}_{\text{DPO}}$ (blue) fails to teach the model to discriminate between preferred and dispreferred responses, as shown by the smallest margin. On the contrary, also optimizing the intermediate task with $\mathcal{L}_{\text{M-DPO}^-}$ further enlarges the margin between responses (green). Furthermore, only optimizing samples where the preference gap between responses is large enough with $\mathcal{L}_{\text{M-DPO}}$ (orange) results in the largest margin, which is 6.6 times of that in $\mathcal{L}_{\text{DPO}}$ at the end of training. The results show that each component in M-DPO contributes to teaching a VideoLLM to better discriminate between preferred and dispreferred responses.

4.4 Qualitative Analysis

In Fig. 2, qualitative results of a baseline (single-turn), and results of a VideoLLM trained with CoTasks and VidChain (CoTasks + M-DPO) are reported. As illustrated, baseline VideoLLM shows inferior performance on DVC, segmenting a video into an overly large number of events (6 predicted vs. 3 ground-truth events), while captions for each event are also highly repetitive ("She clips the cat's other paw and pets him"), revealing the lack of a baseline in the capability of fine-grained video understanding. In contrast, a VideoLLM trained with CoTasks successfully segments a video into three events, while captions for each segment are more distinctive, showing the effectiveness of CoTasks in DVC by decomposing the complex task into multiple sub-tasks. Moreover, a VideoLLM further aligned to metrics with M-DPO produces the best result, where the timestamps predicted and captions generated for each event are closest to ground-truths, demonstrating the efficacy of M-DPO in providing supervision well aligned with the metric.

5 Conclusion

In this paper, we propose a framework named VidChain comprised of the Chain-of-Tasks (CoTasks), and Metric-based Direct Preference Optimization (M-DPO). CoTasks decompose the complicated task into a series of sub-tasks, easing the difficulties in solving the task. M-DPO aligns a VideoLLM with evaluation metrics of sub-tasks, providing supervision aligned with the abilities required for those tasks. Applied on two different VideoLLMs, VidChain enhances fine-grained understanding of models, consistently improving their performance thereby outperforming previous VideoLLMs on multiple fine-grained video understanding benchmarks.

References

Ahn, D.; Choi, Y.; Yu, Y.; Kang, D.; and Choi, J. 2024. Tuning large multimodal models for videos using reinforcement learning from ai feedback. In *ACL*.

Banerjee, S.; and Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL*.

Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; et al. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*.

Cheng, Z.; Leng, S.; Zhang, H.; Xin, Y.; Li, X.; Chen, G.; Zhu, Y.; Zhang, W.; Luo, Z.; Zhao, D.; et al. 2024. VideoLLaMA 2: Advancing Spatial-Temporal Modeling and Audio Understanding in Video-LLMs. arXiv:2406.07476.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. In *NeurIPS*.

Dai, W.; Li, J.; Li, D.; Tiong, A.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; and Hoi, S. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*.

Fujita, S.; Hirao, T.; Kamigaito, H.; Okumura, M.; and Nagata, M. 2020. SODA: Story oriented dense video captioning evaluation framework. In *ECCV*.

Gunjal, A.; Yin, J.; and Bas, E. 2024. Detecting and preventing hallucinations in large vision language models. In *AAAI*.

Huang, B.; Wang, X.; Chen, H.; Song, Z.; and Zhu, W. 2024. Vtimellm: Empower llm to grasp video moments. In *CVPR*.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Carlos Niebles, J. 2017. Dense-captioning events in videos. In *ICCV*.

Lai, X.; Tian, Z.; Chen, Y.; Yang, S.; Peng, X.; and Jia, J. 2024. Step-DPO: Step-wise Preference Optimization for Long-chain Reasoning of LLMs. arXiv:2406.18629.

Li, K.; He, Y.; Wang, Y.; Li, Y.; Wang, W.; Luo, P.; Wang, Y.; Wang, L.; and Qiao, Y. 2023. Videochat: Chat-centric video understanding. arXiv:2305.06355.

Li, K.; Wang, Y.; He, Y.; Li, Y.; Wang, Y.; Liu, Y.; Wang, Z.; Xu, J.; Chen, G.; Luo, P.; et al. 2024. Mvbench: A comprehensive multi-modal video understanding benchmark. In *CVPR*.

Lin, B.; Zhu, B.; Ye, Y.; Ning, M.; Jin, P.; and Yuan, L. 2023. Video-llava: Learning united visual representation by alignment before projection.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024. Improved baselines with visual instruction tuning. In *CVPR*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. In *NeurIPS*.

Maaz, M.; Rasheed, H.; Khan, S.; and Khan, F. S. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Qian, L.; Li, J.; Wu, Y.; Ye, Y.; Fei, H.; Chua, T.-S.; Zhuang, Y.; and Tang, S. 2024. Momentor: Advancing video large language model with fine-grained temporal reasoning. In *ICML*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*.

Ren, S.; Yao, L.; Li, S.; Sun, X.; and Hou, L. 2024. Timechat: A time-sensitive multimodal large language model for long video understanding. In *CVPR*.

Song, F.; Yu, B.; Li, M.; Yu, H.; Huang, F.; Li, Y.; and Wang, H. 2024. Preference ranking optimization for human alignment. In *AAAI*.

Sun, Z.; Shen, S.; Cao, S.; Liu, H.; Li, C.; Shen, Y.; Gan, C.; Gui, L.-Y.; Wang, Y.-X.; Yang, Y.; et al. 2023. Aligning large multimodal models with factually augmented rlhf. arXiv:2309.14525.

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *CVPR*.

Xu, H.; Sharaf, A.; Chen, Y.; Tan, W.; Shen, L.; Van Durme, B.; Murray, K.; and Kim, Y. J. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. In *ICML*.

Yang, A.; Nagrani, A.; Seo, P. H.; Miech, A.; Pont-Tuset, J.; Laptev, I.; Sivic, J.; and Schmid, C. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*.

Ye, Q.; Xu, H.; Ye, J.; Yan, M.; Hu, A.; Liu, H.; Qian, Q.; Zhang, J.; and Huang, F. 2024. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *CVPR*.

Yu, T.; Yao, Y.; Zhang, H.; He, T.; Han, Y.; Cui, G.; Hu, J.; Liu, Z.; Zheng, H.-T.; Sun, M.; et al. 2024. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *CVPR*.

Yuan, L.; Cui, G.; Wang, H.; Ding, N.; Wang, X.; Deng, J.; Shan, B.; Chen, H.; Xie, R.; Lin, Y.; et al. 2024. Advancing llm reasoning generalists with preference trees. arXiv:2404.02078.

Zhang, H.; Li, X.; and Bing, L. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *EMNLP-Demo*.

Zhou, L.; Xu, C.; and Corso, J. 2018. Towards automatic learning of procedures from web instructional videos. In *AAAI*.

Zhu, B.; Lin, B.; Ning, M.; Yan, Y.; Cui, J.; Wang, H.; Pang, Y.; Jiang, W.; Zhang, J.; Li, Z.; et al. 2024. LanguageBind: Extending Video-Language Pretraining to N-modality by Language-based Semantic Alignment. In *ICLR*.

Reproducibility Checklist

This paper:

- Includes a conceptual outline and/or pseudocode description of AI methods introduced (**yes**/partial/no/NA)
- Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (**yes**/no)
- Provides well marked pedagogical references for less-familiare readers to gain background necessary to replicate the paper (**yes**/no)

Does this paper make theoretical contributions? (yes/**no**) If yes, please complete the list below.

- All assumptions and restrictions are stated clearly and formally. (yes/partial/no)
- All novel claims are stated formally (e.g., in theorem statements). (yes/partial/no)
- Proofs of all novel claims are included. (yes/partial/no)
- Proof sketches or intuitions are given for complex and/or novel results. (yes/partial/no)
- Appropriate citations to theoretical tools used are given. (yes/partial/no)
- All theoretical claims are demonstrated empirically to hold. (yes/partial/no/NA)
- All experimental code used to eliminate or disprove claims is included. (yes/no/NA)

Does this paper rely on one or more datasets? (**yes**/no) If yes, please complete the list below.

- A motivation is given for why the experiments are conducted on the selected datasets (**yes**/partial/no/NA)
- All novel datasets introduced in this paper are included in a data appendix. (yes/**partial**/no/NA)
- All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (**yes**/partial/no/NA)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. (**yes**/no/NA)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. (**yes**/partial/no/NA)
- All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing. (yes/partial/no/**NA**)

Does this paper include computational experiments? (**yes**/no)

If yes, please complete the list below.

- Any code required for pre-processing data is included in the appendix. (yes/**partial**/no).
- All source code required for conducting and analyzing the experiments is included in a code appendix. (yes/**partial**/no)
- All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (**yes**/partial/no)
- All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/**partial**/no)
- If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. (**yes**/partial/no/NA)
- This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. (**yes**/partial/no)
- This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. (**yes**/partial/no)
- This paper states the number of algorithm runs used to compute each reported result. (**yes**/no)
- Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. (**yes**/no)
- The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). (yes/partial/**no**)
- This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. (**yes**/partial/no/NA)
- This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. (**yes**/partial/no/NA)