UNICORN: A Unified Causal Video-Oriented Language-Modeling Framework for Temporal Video-Language Tasks

Anonymous ACL submission

Abstract

The great success of large language models has encouraged the development of large multi-modal models, with a focus on image-language interaction. Despite promising results in various image-language downstream tasks, it is still challenging and unclear how to extend the capabilities of these models to the more complex video domain, especially when dealing with explicit temporal signals. To address the problem in existing large multimodal models, in this paper we adopt visual instruction tuning to build a unified causal video-oriented language modeling framework, named UNICORN. Specifically, we collect a comprehensive dataset under the instruction-following format, and instruction-tune the model accordingly. Experimental results demonstrate that without customized training objectives and intensive pre-training, UNICORN can achieve comparable or better performance on established temporal video-language tasks including moment retrieval, video paragraph captioning and dense video captioning. Moreover, the instruction-tuned model can be used to automatically annotate internet videos with temporally-aligned captions. Compared to commonly used ASR captions, we show that training on our generated captions improves the performance of video-language models on both zero-shot and fine-tuning settings. Source code can be found here and will be released upon acceptance.

1 Introduction

Recent breakthroughs in large language models (LLMs) (Ouyang et al., 2022; cha, 2023; OpenAI, 2023; vic, 2023; Touvron et al., 2023a,b) have reignited the enthusiasm about the achievement of artificial general intelligence where a single foundation model can accomplish a large variety of downstream tasks based on human instructions. Towards this ultimate goal, the community has witnessed promising advances in large multimodal models (LMMs) for vision and language (Liu et al., 2023b,a; Wang et al., 2023b; Dai et al., 2023; Bai et al., 2023; Li et al., 2023a; Zhu et al., 2023), the two essential modalities to understand the world. Most of these LMMs follow the pipeline of visual instruction tuning (Liu et al., 2023b) and demonstrate strong capabilities in vision-centric tasks like image classification and object detection (Wang et al., 2023b), and vision-language tasks like image captioning and visual question answering (Dai et al., 2023; Liu et al., 2023b).

Despite impressive results in the image domain, videos, another important data format in the vision modality, are under-explored. In contrast to images, videos have an extra temporal dimension and are much more difficult to process due to increased complexity. Existing approaches either directly apply LMMs trained on image-text pairs (Dai et al., 2023) to the video domain without fine-tuning or develop video-oriented LMMs (Zhang et al., 2023a; Muhammad Maaz and Khan, 2023; Li et al., 2023c) on short trimmed videos. However, such models are limited to handle problems which are less dependent on temporal information like action recognition and video question answering. It still remains unclear how to solve video-language tasks that requires explicitly modeling temporal information, including moment retrieval (Hendricks et al., 2017; Lei et al., 2021), video paragraph captioning (Park et al., 2019), and dense video captioning (Krishna et al., 2017) in one single LMM.

In fact, the inherent disparities among these task formats pose a challenge to the development of such models: moment retrieval requires predicting the temporal location of a moment described by language, paragraph captioning entails to write a coherent story from an untrimmed video, while the goal of dense video captioning is to generate captions and temporal locations for a series of moments simultaneously. These tasks are typically solved individually by specifically-designed models (Lei et al., 2020a; Yang et al., 2023; Lei

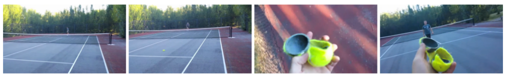
Visual input example, Playing Tennis (34s in total):	
	
Task 1: Moment Retrieval	
Instruction	Please predict start and end time of the following moment: ***He hits the ball over the net several times***. The output format should be <start><end>.
Response	<16><48>
Task 2: Video Paragraph Captioning	
Instruction	Provide a detailed description of the video, capturing its key moments.
Response	A man is bouncing a tennis ball on an outdoor court. He hits the ball over the net several times. The balls roll over to the opposing fence, broken in half from the impact.

Table 1: Example of instruction-following data. The response of moment retrieval is computed by time tokenization for the window [7.7s, 22.1s] with 75 bins.

et al., 2021; Lin et al., 2023). While attempts have been made to unify these temporal video-language tasks (Wang et al., 2023a; Yan et al., 2023), separate modules and training objectives tailored for each task are involved in these methods, making them complicated in both training and inference.

To address the above challenge, we propose a **UNI**fied **C**ausal vide**O**-o**R**iented la**N**guage modeling framework (**UNICORN**) that unifies the tasks as a simple yet generic language modeling problem. For moment retrieval and video paragraph captioning, we convert original training datasets into corresponding instruction-following formats, as shown in Table 1. In particular, inspired by previous efforts in discretizing bounding box coordinates (Chen et al., 2022; Peng et al., 2023; Zhang et al., 2023b), our approach represents the continuous event boundaries as a sequence of discrete time tokens and processes them similarly as language tokens. On a range of datasets and tasks, we show that this unified approach achieve comparable or better performance over previous methods.

On the other hand, the development of large video-language models is hindered by the lack of semantically- and temporally-aligned video-text pairs, an issue unique to the video domain. As pointed out in (Han et al., 2022), the models pre-trained on commonly-used noisy datasets such as HowTo100M (Miech et al., 2019) and YT-Temporal-1B (Zellers et al., 2022) suffer from the misalignment between videos and ASR captions severely. Thanks to the generalization ability of LMMs, our UNICORN can be leveraged to automatically generate captions for internet videos. We demonstrate that **qualitatively** the generated captions are better semantically- and temporally-aligned with the videos than the original ASR captions, and **quantitatively** incorporating our generated captions in either instruction-tuning for moment retrieval or end-to-end video representation learning leads to significant performance gains.

Our contributions are threefold: **(1)** We propose UNICORN, a simple and generic framework that unifies various temporal video-language tasks via language modeling; **(2)** Our approach achieves comparable or better performance to state-of-the-art methods on multiple downstream tasks, including moment retrieval, video paragraph captioning, and dense video captioning; **(3)** Compared to existing captions, those automatically generated by our method have shown to be better aligned with the videos, both semantically and temporally. Empirically, the generated captions have demonstrated to improve performance of models trained on them. Our automatic annotation pipeline is useful for empowering the development of future LMMs.

2 Related Work

Large Multimodal Models. Large language models are taking the world by storm with their incredible capabilities to answer questions in a coherent and informative way aligned with human instructions (cha, 2023; Ouyang et al., 2022; vic, 2023; OpenAI, 2023; Touvron et al., 2023a,b). The universality and generalization of LLMs make it potential to unlock the door to a foundation general-purpose model. Towards this goal, a variety of large multimodal models are emerging to bridge different modalities, in particular vision and language (Liu et al., 2023b,a; Wang et al., 2023b; Dai et al., 2023; Bai et al., 2023; Li et al., 2023a; Zhu et al., 2023). Such LMMs adopt the pipeline of visual instruction tuning (Liu et al., 2023b) by converting original datasets into the instruction-following format and casting traditional vision problems as a language modeling task. For instance, LLaVa (Liu et al., 2023b) generates multimodal language-image instructional data using GPT-4 (OpenAI, 2023) and develops an LMM connecting a pre-trained image encoder and a pre-trained large language model to deal with vision-language tasks. InstructBLIP (Dai et al., 2023) enlarges the task coverage by gathering 26 publicly available datasets and proposes an instruction-aware visual feature extraction process. These models achieve the state-of-the-art performance on numerous downstream tasks, ranging from vision-centric ones such as image classification and object detection to vision-language ones such as image captioning and visual reasoning. Despite efforts in understanding images, few attempts have been made for video-language tasks due to additional complexity. Thus, in this paper we study how to model the interaction between long untrimmed videos and captions from the perspective of language modeling.

Video-Language Modeling. Video-language tasks have been widely studied, especially these requires specific temporal modeling, such as moment retrieval (Lei et al., 2021; Lin et al., 2023; Mun et al., 2020; Zeng et al., 2020), video paragraph captioning (Lei et al., 2020a; Park et al., 2019; Yang et al., 2023; Wang et al., 2021), and dense video captioning (Krishna et al., 2017; Yang et al., 2023; Wang et al., 2021). Some methods (Lin et al., 2023; Yan et al., 2023; Wang et al., 2023a; Li et al., 2022) pre-train a model on large-scale corpus to generate latent video and language representations, which can be then adapted to different downstream tasks. This line of work typically requires elaborate architectural designs and multiple training objectives tailored for each target task. In contrast, we propose a more elegant unified framework to integrate various temporal video-language tasks into a simple yet generic language modeling problem. Compared with existing video-oriented LMMs targeting at short video clips (Li et al., 2023c; Zhang et al., 2023a; Muhammad Maaz and Khan, 2023), UNICORN attaches more attention to long untrimmed videos. The most relevant method to UNICORN is Vid2Seq (Yang et al., 2023), which also formulates dense video captioning as language modeling. However, it should be emphasized that Vid2Seq depends heavily on video-language pre-training and is unable to handle tasks other than captioning. On the contrary, by visual instruction tuning on high quality datasets, UNICORN demonstrates superior performance on a series of video-language tasks without intensive pre-training. Moreover, our method can be applied towards noisy video datasets to generate better-aligned captions.

3 Method

In this section, we introduce our unified framework UNICORN in detail. We start by discussing how to transform the original datasets for different downstream tasks into the general instruction-following format in Section 3.1. Then in Section 3.2, we describe the model architecture designed for video-language interaction. In Section 3.3, we present the training pipeline of UNICORN including datasets and training objective. Finally in Section 3.3, we demonstrate how to conduct inference with the obtained model on downstream tasks together with the process to generate captions for noisy datasets.

3.1 Instruction-Following Data Generation

As the ultimate goal is to unify various temporal video-language tasks, we cast moment retrieval and video paragraph captioning datasets into a common instruction following format. For dense video captioning, it can be regarded as a two-stage procedure of paragraphing captioning and moment retrieval and thus no specific training data are required. We provide details in following sections.

Moment Retrieval In moment retrieval (MR) (Hendricks et al., 2017; Gao et al., 2017; Krishna et al., 2017; Lei et al., 2020b, 2021), a continuous time window is predicted given an untrimmed video and a language moment query. With the task definition, an example instruction can be: "**Please predict start and end time of the following moment: {target}**", where **{target}** is replaced by the specific query. We curate a template instruction list in Appendix B, to explicitly teach the underlying model the concepts of the task and the objective.

A key challenge here is how to generate output sequences to represent moment locations. To reduce the exploration space for more controllable predictions, we follow previous sequence generation strategies for such continuous values (Chen et al., 2022; Peng et al., 2023; Yang et al., 2023; Wang et al., 2023b; Chen et al., 2023), and discretize the timestamp t in a d-s long video into an integer in $\{0, 1, \ldots, N_{\text{bin}} - 1\}$ with N_{bin} equally-spaced bins by $\lfloor t \times N_{\text{bin}} \rfloor / d$. Moreover, since recent LLMs exhibit surprising performance in mathematical reasoning, we use the original vocabulary without extra time tokens, which in turn reduces the number of trainable parameters and avoids pre-training to re-acquire the ability to reason about numbers. Meanwhile, to distinguish our discrete relative timestamps from other numerical expressions such as "5 apples", we enclose the timestamp values by "<start><end>" where start and end are replaced by corresponding converted timestamps. For instance, the moment in Table 1 starting at 7.7s and ending at 22.1s within a 34s-long video

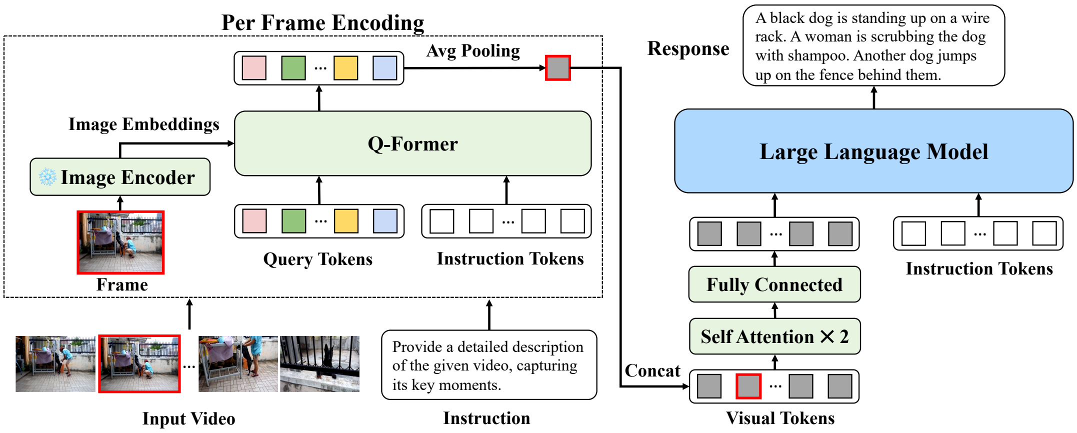


Figure 1: UNICORN framework using video paragraph captioning as an example. We encode each video frame separately and concatenate their resulting visual tokens to represent the video. We highlight the encoding process of one frame in red. All modules are instruction-tuned with the language modeling loss except the image encoder.

is transformed into the desired output sequence "<16><48>" after our proposed time tokenization with 75 bins. To make output predictions consistent in format, we append a language constraint to our instruction: "**The output format should be <start><end>.**" For a moment query associated with multiple time windows, we regard each query-location pair as an individual data sample.

Video Paragraph Captioning The task of video paragraph captioning (VPC) (Park et al., 2019; Lei et al., 2020a) aims at generating a set of coherent sentences to describe an untrimmed video that contains several events. While previous pipelines (Park et al., 2019; Lei et al., 2020a) segment the video into multiple clips from ground-truth event boundary proposals, our method takes as input frames sampled from the whole video together with the instruction "**Provide a detailed description of the given video, capturing its key moments.**". We generate a diverse template set in Appendix B to reduce overfitting and strengthen the understanding of the task. We leverage the paragraph caption of the target video as the prediction.

Dense Video Captioning The goal of dense video captioning (DVC) (Krishna et al., 2017) is to generate multiple corresponding captions for a series of events together with their temporal locations from the untrimmed video. It is much harder than moment retrieval and paragraph captioning since it requires predicting events and their timestamps simultaneously. The most straightforward way to convert the task into the instruction-following format is to construct a sequence with both events and locations given a specific input prompt. However, design choices such as event serialization (e.g., chronological or random) and where to insert time windows might affect the performance significantly (Chen et al., 2022; Yang et al., 2023). Furthermore, the training of such models is challenged by the longer input sequence with both timestamps and event descriptions. It also takes extra computational costs to learn redundant information from moment retrieval and video paragraph captioning again. Considering the inherent property of DVC, we find that it can be naturally decomposed into a two-stage procedure of video paragraph captioning followed by moment retrieval. Thus, no additional training data are required and this task can be addressed at inference-time by the model instruction-tuned on two tasks above, with more details in Section 3.3.

3.2 Model Architecture

To bridge together video frames and natural language instructions as the ultimate input sequence, we propose a large multimodal language model, demonstrated in Figure 1. Specifically, a sequence of visual tokens are obtained by feeding frames and the corresponding instruction from an untrimmed video into our per-frame encoding module. Visual tokens are then processed by a projection layer to the same latent space as the large language model (LLM). The LLM takes the concatenation of visual tokens and instruction tokens as input and generates the desired output given different task instructions.

Compared with image-language interaction, few attempts have been made in the video domain due to increased complexity. However, using image encoders to conduct per-frame encoding for videos by brute force will lead to an extremely long sequence of visual tokens proportional to the number of frames. On the other hand, a completely new encoder might require a considerable amount of training to align modalities of vision and language again.

To strike a balance between two aforementioned issues, we resort to the recently proposed InstructBLIP (Dai et al., 2023) and make some adaptions on the Q-Former module to handle the video input. In detail, our method first extracts n_q visual tokens from each frame using the frame-based encoding of the original Q-Former. For efficiency, we then apply average pooling in a frame-wise manner, which results in one token for each frame. Given a video of N frames, these N tokens are further processed by a module with two self-attention layers to integrate temporal information. Our design maintains a reasonable length of visual tokens for instruction tuning and takes advantage of pre-trained LLMs for feature alignment between the two modalities.

3.3 Training and Inference

With the data in instruction-following format, we now present a unified framework of instruction tuning on various downstream tasks.

Training. The instruction-following format makes it feasible to train the model to predict next tokens with an auto-regressive language modeling loss. Given input video frames $X=\{x_i\}_{i=1}^{N}$ and task instruction $Y=\{y_j\}_{j=1}^{M}$, we maximize the log likelihood of the output sequence $Z=\{z_k\}_{k=1}^{L}$: $\max \sum_{k=1}^{L} \log p_\theta(z_k|X, Y, z_{1:k-1})$, where L is the output sequence length, p_θ is the output probability distribution over the LLM vocabulary given model parameters θ. We finetune the whole model except the image encoder using LoRA (Hu et al., 2022).

Inference. For moment retrieval and paragraph captioning, we prompt the instruction tuned model using corresponding task instructions to generate responses via beam search. For dense video captioning, we divide it into two stages, where the model first generates a paragraph caption, and then temporally locate each sentence in the paragraph with moment retrieval task instruction.

4 Experiments

In this section, we evaluate UNICORN comprehensively against state-of-the-art methods to show its effectiveness. We first introduce experimental setup in Section 4.1. Then we present results on downstream tasks including moment retrieval, video paragraph captioning and dense video captioning in Section 4.2. Ablation studies are conducted in Section 4.3 for better understanding of our designs. Finally, in Section 4.4 we investigate the quality of the automatic annotation generated by UNICORN on HowTo100M.

4.1 Experimental Setups

Architecture. The backbone of our video encoding module is adapted from InstructBLIP (Dai et al., 2023). Specifically, we implement the video encoder with the same image encoder (ViT-G/14) (Fang et al., 2023), Q-Former with 32 learnable query embeddings and a fully-connected projection layer as the original InstructBLIP structure, plus a temporal modeling module with 2 self-attention layers. For the language side, we select Vicuna-7B (vic, 2023), a publicly available LLM fine-tuned from LLaMa (Touvron et al., 2023a).

Datasets. Rather than intensive pre-training on a large scale noisy dataset without annotations, we directly fine-tune our model on a comprehensive set of publicly available video-language datasets, including QVHighlights (Lei et al., 2021), Charades-STA (Gao et al., 2017), ActivityNet Captions (Krishna et al., 2017), and YouCook2 (Zhou et al., 2018a). The collection covers various domains with different length distributions. More details about datasets are included in Appendix C.

Implementation details. We adopt LAVIS (Li et al., 2023b) under BSD 3-Clause License to run all the experiments and our usage is compatible with its license. The model is instruction tuned for 5 epochs with a batch size of 32. We randomly sample a task at a time based on data size. We use AdamW (Loshchilov and Hutter, 2019) with β_1=0.9, β_2=0.999, and weight decay 0.05 for optimization. The learning rate is warmuped from 10^{-6} to 10^{-4} in the first epoch, followed by a cosine decay with a minimum of 10^{-5}. We freeze the image encoder and fine-tune the rest of the model, with LoRA applied on the LLM. There are around 243M trainable parameters. UNICORN is trained with 8 NVIDIA A100 (80G) GPUs in 12 hours.

Evaluation. For moment retrieval, we evaluate on QVHighlights, Charades-STA, and ActivityNet Captions. We report the standard metrics Recall at 1 under temporal Intersection over Union (IoU) thresholds of 0.5 and 0.7, abbreviated as R@0.5 and R@0.7. Besides, we use the average mAP over IoU thresholds [0.5:0.05:0.95] on QVHighlights with multiple ground-truth segments for one moment, and mean IoU (mIoU) for the other two datasets. For video paragraph captioning, we use commonly-adopted metrics CIDEr (Vedantam et al., 2015) (C) and METEOR (Banerjee and Lavie, 2005) (M) and report results on YouCook2 and ActivityNet Captions. As to dense video cap-

Method	QVHighlights			Charades-STA			ActivityNet Captions		
	R@0.5	R@0.7	mAP avg	R@0.5	R@0.7	mIoU	R@0.5	R@0.7	mIoU
LGI (Mun et al., 2020)	—	—	—	59.5	35.5	51.4	41.5	23.1	41.1
2D TAN (Zhang et al., 2020b)	—	—	—	46.0	27.5	41.2	44.5	26.5	—
VSLNet (Zhang et al., 2020a)	—	—	—	42.7	24.1	41.6	43.2	26.2	43.2
MDETR (Lei et al., 2021)	59.8	40.3	36.1	52.1	30.6	45.5	—	—	—
GVL (Wang et al., 2023a)	—	—	—	—	—	—	**48.9**	27.2	46.4
UnLoc (Yan et al., 2023)	64.5	48.8	—	58.1	35.4	—	48.0	29.7	—
UniVTG (Lin et al., 2023)	58.9	40.9	35.5	58.0	35.6	50.1	—	—	—
UniVTG, PT (Lin et al., 2023)	65.4	50.1	43.6	60.2	38.5	52.2	—	—	—
UNICORN	**68.4**	**51.9**	**45.0**	**69.0**	**45.6**	**58.9**	48.4	**29.8**	**47.1**

Table 2: Moment retrieval on QVHighlights (*test*), Charades-STA (*test*), and ActivitityNet Captions (*val_2*). We **bold** the best, underline the second-best.

tioning, we follow the existing protocol (Krishna et al., 2017) to compute captioning metrics over the matched pairs between generated sentences and the ground truth. SODA_c (Fujita et al., 2020) (S) is also used to measure the temporal coherence for a set of captions. This task is evaluated on YouCook2 and ActivityNet Captions as well.

4.2 Results

We evaluate our instruction-tuned model on three video-language tasks: moment retrieval, video paragraph captioning, and dense video captioning. Note that all results are obtained from one shared model and different tasks are addressed by changing the prompting instructions *at inference time* only.

Moment retrieval. In Table 2, our method is compared with state-of-the-art algorithms for this task on three representative datasets, QVHighlights (Lei et al., 2021), Charades-STA (Gao et al., 2017), and ActivityNet Captions (Krishna et al., 2017). It can be observed that our method achieves comparable (mostly better) performance on all three datasets. In particular, on QVHighlights we achieve 68.4, 51.9, and 45.0 for R@0.5, R@0.7 and average mAP respectively, improving the best-performing baseline UniVTG with pre-training substantially by +3.0, +1.8 and +1.4. In contrast to complicated designs such as a localization loss in previous approaches, we remove most of the specification and only use a generic language modeling loss: UNICORN is mainly based on the intuition that if a model knows about where the moment is, we just need to teach it how to read the location out. In summary, UNICORN makes minimal assumptions on the task yet accomplishes it with superior performance.

Video paragraph captioning. Table 3 shows the video paragraph captioning results. In UNICORN, we consider this task as a general captioning problem. Without any customized training objectives or prior knowledge on the input such as ground-truth event proposals as in previous methods (Park et al., 2019; Lei et al., 2020a), our method demonstrates outstanding performance over other baselines under both settings of ground truth or learned proposals. It further showcases the strong adaptation of LMMs to downstream tasks through instruction tuning with high-quality instruction-following data.

Method	Backbone	YouCook2		ActivityNet	
		C	M	C	M
With GT Proposals					
VTransformer (Zhou et al., 2018b)	V (ResNet-200) + F	32.3	15.7	22.2	15.6
Transformer-XL (Dai et al., 2019)	V (ResNet-200) + F	26.4	14.8	21.7	15.1
MART (Lei et al., 2020a)	V (ResNet-200) + F	35.7	15.9	23.4	15.7
GVDSup (Zhou et al., 2019)	V (ResNet-101) + F + O	—	—	22.9	16.4
AdvInf (Park et al., 2019)	V (ResNet-101) + F + O	—	—	21.0	16.6
PDVC (Wang et al., 2021)	V + F (TSN)	—	—	27.3	15.9
With Learned Proposals					
MFT (Xiong et al., 2018)	V + F (TSN)	—	—	19.1	14.7
PDVC (Wang et al., 2021)	V + F (TSN)	—	—	20.5	15.8
PDVC (Wang et al., 2021)	V (CLIP)	—	—	23.6	15.9
TDPC (Song et al., 2021)	V (ResNet-200) + F	—	—	26.5	15.6
Vid2Seq (Yang et al., 2023)	V (CLIP)	—	—	28.0	17.0
GVL (Wang et al., 2023a)	V (TSN)	—	—	26.0	16.3
UNICORN	V (CLIP)	**37.8**	**18.3**	**34.8**	**17.3**

Table 3: Video paragraph captioning results on YouCook2 (*val*) and ActivityNet Captions (*ae-test*). V/F/O refers to visual/flow/object features.

Method	Backbone	YouCook2			ActivityNet		
		S	C	M	S	C	M
MT (Zhou et al., 2018b)	TSN	—	6.1	3.2	—	9.3	5.0
ECHR (Wang et al., 2020)	C3D	—	—	3.8	3.2	14.7	7.2
PDVC (Wang et al., 2021)	TSN	4.4	22.7	4.7	5.4	29.0	8.0
PDVC (Wang et al., 2021)	CLIP	4.9	28.9	5.7	6.0	29.3	7.6
UEDVC (Zhang et al., 2022)	TSN	—	—	—	5.5	26.9	7.3
E2ESG (Zhu et al., 2022)	C3D	—	25.0	3.5	—	—	—
Vid2Seq (Yang et al., 2023)	CLIP	**5.7**	25.3	—	5.9	30.2	8.5
GVL (Wang et al., 2023a)	TSN	4.9	26.5	5.0	6.2	32.8	8.5
UNICORN	CLIP	**5.7**	**37.0**	**7.7**	**6.3**	**35.4**	**9.2**

Table 4: Results of DVC on YouCook2 (*val*) and ActivityNet Captions (*val_1* and *val_2*).

Dense video captioning. We generate dense video captions following the procedure in Section 3.1 and evaluate the performance in Table 4. It can be observed that our method takes the lead among the compared approaches, including Vid2Seq which leverages language models to predict captions and timestamps simultaneously. These promising results also validate our divide-and-conquer strategy for dense video captioning. Such an inference design makes the training more efficient without learning on redundant and lengthy DVC data again while still achieving competitive results.

4.3 Ablation Studies

We conduct ablation studies to analyze effects of the key components in UNICORN, including training strategies, the choice of time tokens, and various model designs. We evaluate on QVHighlights (*val*) for moment retrieval and ActivityNet Captions (*ae-test*) for video paragraph captioning. Additional analysis including base model selection can be found at Appendix D.

Training strategies. We study the effects of training strategies for UNICORN. Specifically, three strategies are considered: single-task & single dataset, single task & multi-dataset, and multi-

(a) Comparison of training strategies.

Training Setup	QVHighlights			ActivityNet	
	R@0.5	R@0.7	mAP	C	M
Single-task, single-dataset	66.3	51.5	42.8	33.6	16.4
Single-task, multi-dataset	68.2	52.3	44.8	34.6	16.9
Multi-task, multi-dataset	**69.5**	**54.4**	**45.3**	**34.8**	**17.3**

(b) The number of frames.

#frames	QVHighlights			ActivityNet	
	R@0.5	R@0.7	mAP	C	M
25	61.5	37.4	35.0	33.4	16.9
50	65.4	47.9	41.4	34.5	17.0
75	**69.5**	**54.4**	**45.3**	**34.8**	**17.3**
100	67.8	52.8	44.7	34.6	**17.3**

(c) LoRA & temporal modeling.

LoRA	Temporal modeling	QVHighlights			ActivityNet	
		R@0.5	R@0.7	mAP	C	M
✗	✗	60.6	36.4	33.2	23.0	16.0
✓	✗	66.7	49.2	39.8	34.4	17.2
✗	✓	65.5	47.0	40.4	27.6	16.8
✓	✓	**69.5**	**54.4**	**45.3**	**34.8**	**17.3**

Table 5: Ablation studies on training strategies and model designs of LoRA and temporal modeling.

task & multi-dataset. For the single-task version, we fine-tune two separate models with corresponding instructions tailored for moment retrieval and video paragraph captioning respectively, and select one representative dataset for each task for evaluation. For the single-dataset version, we train only on the training split of the evaluation dataset (i.e., QVHighlights for moment retrieval and ActivityNet Captions for video paragraph captioning)

We report detailed results in Table 5a. By introducing datasets from different domains for the same task, we can improve the model's capability on the single dataset. Besides, in contrast to traditional multi-task training strategies, instruction tuning on various descriptions works as a unified approach to integrate different tasks and can even boost the performance from understanding a video from multiple perspectives. Meanwhile, it is more convenient to store only one model to accomplish distinct tasks, which narrows the gap from constructing a general-purpose foundation model.

Time tokens	R@0.5	R@0.7	mAP
Dedicated	64.1	48.2	40.7
Original vocab	**66.3**	**51.5**	**42.8**

Table 6: Different time token.

Time tokens. We can either introduce new dedicated time tokens or directly use the digits in the original vocabulary to represent time. We investigate the impact of the two strategies in the single-task, single-dataset setup on QVHighlights in Table 6. We observed that the original vocabulary performs better than new dedicated tokens, which indicates the knowledge of digits in LLM can be readily transferred to our tasks. Meanwhile, new tokens would introduce extra training overheads and increase the number of trainable parameters by 262M, more than double of the original value, see details in Appendix D.

Number of frames. By default, we evenly sample 75 frames from a video as model inputs. In Table 5b, we study the impact with #frames of 25, 50, 75, and 100. The performance generally improves when we adopt more frames while it saturates or even gets worse around 100 frames. Since the videos in the datasets we studied are usually not very long (*e.g.*, videos in QVHighlights are on average 150 seconds long), we hypothesize that 75 frames are enough to cover the semantic information needed for the tasks. We report more results about #frames in Appendix D.

LoRA. We use a parameter-efficient fine-tuning method LoRA to fine-tune the LLM of UNICORN. In Table 5c, LoRA has been proven effective in boosting performance for downstream tasks (row 1 *vs.* 2 and row 3 *vs.* 4). It is expected that frozen LLM would not work properly as we have assigned new meanings to original digit tokens to represent discrete time bins, and LoRA training mitigates the issue without tuning the whole LLM intensively.

Temporal modeling. Since our model is adapted from image-based InstructBLIP, we include an additional module with self-attention layers to incorporate temporal information for videos in Figure 3. As shown in Table 5c, when temporal modeling is enabled from average pooling to self-attention interaction (row 1 *vs.* 3 and row 2 *vs.* 4), there is substantial improvement in moment retrieval and paragraph captioning, indicating the necessity of this module for temporal video-language tasks.

4.4 Auto Annotation of HowTo100M

Thanks to the generalization of LMMs, the model to handle temporal video-language tasks can be deployed on unseen public internet videos such as HowTo100M (Miech et al., 2019). These videos are paired with auto speech recognition (ASR) transcripts, a majority of which are not visually and temporally aligned (Miech et al., 2020; Tang et al., 2021; Han et al., 2022). Since our model is capable of generating dense captions, it is promising to leverage UNICORN for annotating the dataset automatically. We use our trained model to densely caption a subset of 240K videos from HowTo100M (Han et al., 2022) and denote the dataset as HTM-UNICORN. We anonymize names with their pronouns and prompt the model not to generate offensive responses. We compare it with two variants with the same set of videos, HTM-ASR (Miech et al., 2019) with original ASR transcripts, and HTM-AA (Han et al., 2022) which has been aligned temporally via an automated process. Note that UNICORN can output diverse cap-

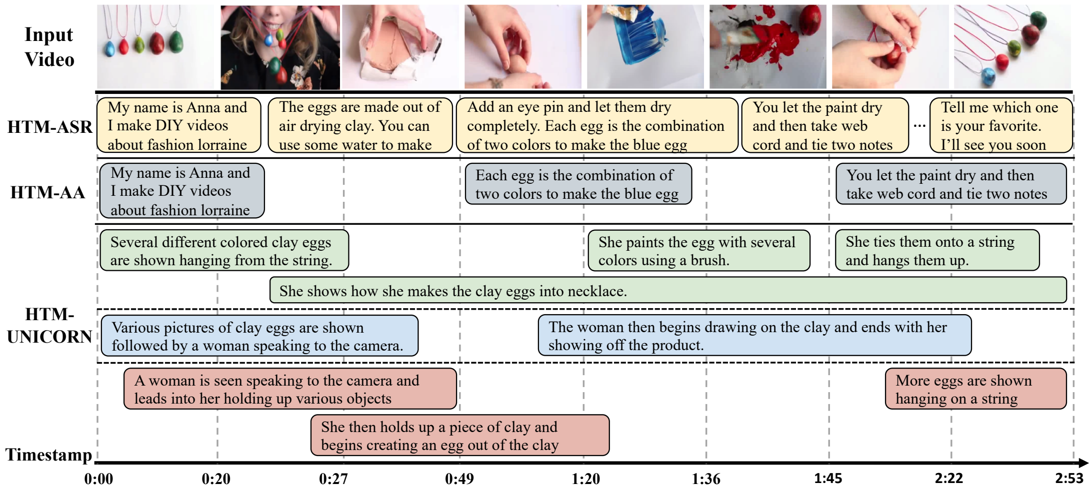


Figure 2: Comparison among captions from HTM-ASR, HTM-AA, and UNICORN respectively. For HTM-UNICORN, we show three sets of generated captions via beam search, coded with different colors.

Dataset	#queries	Zero-shot			Fine-tuning		
		R@0.5	R@0.7	mAP	R@0.5	R@0.7	mAP
InstructBLIP	—	—	—	—	66.3	51.5	42.8
HTM-ASR (Miech et al., 2019)	5.0M	7.7	2.8	1.9	63.5	48.3	40.3
HTM-AA (Han et al., 2022)	3.3M	13.0	4.8	3.5	65.8	50.6	42.1
HTM-UNICORN	690K	44.2	26.4	26.0	68.9	53.6	45.0
HTM-UNICORN ×2	1.4M	47.5	30.8	29.9	69.5	54.0	45.2
HTM-UNICORN ×3	2.1M	**50.0**	**32.7**	**30.4**	**70.2**	**54.6**	**45.5**

Table 7: Zero-shot and fine-tuning moment retrieval evaluation on QVHighlights (*val*). HTM-UNICORN $\times n$ indicates we generated n sets of captions for a video.

tions using beam search (Vijayakumar et al., 2016), which can increase the training data size and as a result improve model performance with more data.

In Figure 2, we present an qualitative comparison of three variants. Our HTM-UNICORN is the best aligned with the input video both visually and temporally, compared with HTM-ASR and HTM-AA. In addition, captions from different sets can complement each other, leading to more comprehensive descriptions of the video. Quantitatively, we use three HowTo100M variants to pre-train the model for moment retrieval, and evaluate on QVHighlights under zero-shot and fine-tuning settings. We convert these datasets into the instruction-following format described in Section 3.1, and train the model from the same initialization. In Table 7, we observe that our automatically annotated HTM achieves superior zero-shot performance, which shows the better alignment of moments and timestamps. For fine-tuning, we notice that performance even degrades when pre-trained on HTM-ASR and HTM-AA, potentially due to data noise, while the model pre-trained on HTM-UNICORN outperforms other variants, reflecting the high quality of the generated dataset.

Besides, we follow (Han et al., 2022) to conduct end-to-end representation learning with an InfoNCE loss (Miech et al., 2020). After contrastive pre-training, we evaluate video representations by

PT Dataset	Backbone	UCF101	HMDB51	K400
HTM-ASR (Miech et al., 2020)	S3D	82.1	55.2	55.7
HTM-AA (Han et al., 2022)	S3D	83.2	56.7	56.2
HTM-UNICORN	S3D	**84.1**	**57.7**	**56.6**

Table 8: Linear probing accuracy for action recognition.

linear probing on three action recognition datasets, UCF101 (Soomro et al., 2012), HMDB51 (Kuehne et al., 2011), and Kinetics-400 (K400) (Kay et al., 2017) in Table 8. UNICORN achieves the highest accuracy on all three datasets, which again demonstrates the best quality of our generated captions.

Captions generated from our automated annotation pipeline has shown to be better than noisy web data both qualitatively and quantitatively. As data quality and quantity are crucial for the performance of large models (Zhou et al., 2024; Ji et al., 2023; Liu et al., 2023a), we hope such a pipeline could be useful for empowering the development of future large multimodal models.

5 Conclusion

In this paper, we propose a unified causal video-oriented language modeling framework UNICORN to address temporal video-language tasks. By fine-tuning on instruction-following data constructed from existing datasets, our model achieves outstanding performance on various downstream tasks including moment retrieval, video paragraph captioning and dense video captioning. We further show that UNICORN can be leveraged in automatic annotation on internet videos such as HowTo100M for semantically- and temporally-aligned captions. These captions can be used to improve video-language model performance against ASR ones. In conclusion, UNICORN paves the way towards a general-purpose foundation model that explicitly considers temporal information.

6 Limitations

Currently, UNICORN is good at localizing an event which only appears once in the video, but would be confused when an event happens more than once. This is due to the training data mostly have events appearing once. Future work can be collecting data with events that appear more than once to improve models' ability on these scenarios.

References

2023. Chatgpt. https://openai.com/blog/chatgpt.

2023. Vicuna. https://github.com/lm-sys/FastChat.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. 2023. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*.

Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. 2022. Pix2seq: A language modeling framework for object detection. In *ICLR*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. 2023. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*.

Soichiro Fujita, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2020. Soda: Story oriented dense video captioning evaluation framework. In *ECCV*.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *ICCV*.

Tengda Han, Weidi Xie, and Andrew Zisserman. 2022. Temporal alignment networks for long-term video. In *CVPR*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *ICCV*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.

Yunjie Ji, Yong Deng, Yan Gong, Yiping Peng, Qiang Niu, Lei Zhang, Baochang Ma, and Xiangang Li. 2023. Exploring the impact of instruction data scaling on large language models: An empirical study on real-world use cases. *arXiv preprint arXiv:2303.14742*.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *CVPR*.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. 2011. Hmdb: a large video database for human motion recognition. In *ICCV*.

Jie Lei, Tamara L Berg, and Mohit Bansal. 2021. Detecting moments and highlights in videos via natural language queries. In *NeurIPS*.

Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara Berg, and Mohit Bansal. 2020a. Mart: Memory-augmented recurrent transformer for coherent video paragraph captioning. In *ACL*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020b. Tvr: A large-scale dataset for video-subtitle moment retrieval. In *ECCV*.

Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Fanyi Pu, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2023a. Mimic-it: Multi-modal in-context instruction tuning. *arXiv preprint arXiv:2306.05425*.

Dongxu Li, Junnan Li, Hung Le, Guangsen Wang, Silvio Savarese, and Steven C.H. Hoi. 2023b. LAVIS: A one-stop library for language-vision intelligence. In *ACL*.

Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. 2022. Align and prompt: Video-and-language pre-training with entity prompts. In *CVPR*.

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

Kevin Qinghong Lin, Pengchuan Zhang, Joya Chen, Shraman Pramanick, Difei Gao, Alex Jinpeng Wang, Rui Yan, and Mike Zheng Shou. 2023. Univtg: Towards unified video-language temporal grounding. In *ICCV*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *NeurIPS*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*.

Salman Khan Muhammad Maaz, Hanoona Rasheed and Fahad Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *ArXiv 2306.05424*.

Jonghwan Mun, Minsu Cho, and Bohyung Han. 2020. Local-global video-text interactions for temporal grounding. In *CVPR*.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. In *NeurIPS*.

Jae Sung Park, Marcus Rohrbach, Trevor Darrell, and Anna Rohrbach. 2019. Adversarial inference for multi-sentence video description. In *CVPR*.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*.

Yuqing Song, Shizhe Chen, and Qin Jin. 2021. Towards diverse paragraph captioning for untrimmed videos. In *CVPR*.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Zineng Tang, Jie Lei, and Mohit Bansal. 2021. DeCEMBERT: Learning from noisy instructional videos via dense captions and entropy minimization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

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

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Teng Wang, Jinrui Zhang, Feng Zheng, Wenhao Jiang, Ran Cheng, and Ping Luo. 2023a. Learning grounded vision-language representation for versatile understanding in untrimmed videos. *arXiv preprint arXiv:2303.06378*.

Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. 2021. End-to-end dense video captioning with parallel decoding. In *ICCV*.

Teng Wang, Huicheng Zheng, Mingjing Yu, Qian Tian, and Haifeng Hu. 2020. Event-centric hierarchical representation for dense video captioning. *IEEE Transactions on Circuits and Systems for Video Technology*.

Wenhai Wang, Zhe Chen, Xiaokang Chen, Jiannan Wu, Xizhou Zhu, Gang Zeng, Ping Luo, Tong Lu, Jie Zhou, Yu Qiao, et al. 2023b. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. *arXiv preprint arXiv:2305.11175*.

Yilei Xiong, Bo Dai, and Dahua Lin. 2018. Move forward and tell: A progressive generator of video descriptions. In *ECCV*.

Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. 2023. Unloc: A unified framework for video localization tasks. In *ICCV*.

Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef

Sivic, and Cordelia Schmid. 2023. Vid2seq: Large-scale pretraining of a visual language model for dense video captioning. In *CVPR*.

Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. 2022. Merlot reserve: Neural script knowledge through vision and language and sound. In *CVPR*.

Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. 2020. Dense regression network for video grounding. In *CVPR*.

Hang Zhang, Xin Li, and Lidong Bing. 2023a. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.

Hao Zhang, Aixin Sun, Wei Jing, and Joey Tianyi Zhou. 2020a. Span-based localizing network for natural language video localization. In *ACL*.

Qi Zhang, Yuqing Song, and Qin Jin. 2022. Unifying event detection and captioning as sequence generation via pre-training. In *ECCV*.

Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. 2023b. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*.

Songyang Zhang, Houwen Peng, Jianlong Fu, and Jiebo Luo. 2020b. Learning 2d temporal adjacent networks for moment localization with natural language. In *AAAI*.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *NeurIPS*, 36.

Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. 2019. Grounded video description. In *CVPR*.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018a. Towards automatic learning of procedures from web instructional videos. In *AAAI*.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018b. End-to-end dense video captioning with masked transformer. In *CVPR*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Wanrong Zhu, Bo Pang, Ashish V Thapliyal, William Yang Wang, and Radu Soricut. 2022. End-to-end dense video captioning as sequence generation. In *COLING*.

A Societal Impact

Similar to many data-driven methods, the predictions from our model might be inaccurate and biased towards the distribution of data on which it is trained on. Therefore, users should not completely rely on the model in real-world scenarios.

B Instruction Templates

We provide the list of instruction templates for moment retrieval and video paragraph captioning respectively in Table 9 and Table 10.

- "Please predict start and end time of the following moment."
- "Can you tell me the time window of this event?"
- "What is the location of the moment?"

Table 9: The list of instructions for moment retrieval.

- "Provide a detailed description of the given video, capturing its key moments."
- "Describe the following video in detail, including the actions and scenes."
- "Clarify the contents of the displayed video with great detail, focusing on its progression."
- "Offer a thorough analysis of the video, discussing its various elements and storyline."

Table 10: The list of instructions for video paragraph captioning.

C Datasets

In this section, we present more details about datasets used for both instruction-tuning and evaluation. An overview of statistics of training data is presented in Table 11. We mix all samples of the same task across datasets and obtain two large training sets: one for moment retrieval S_{MR} with $|S_{\text{MR}}|$=71829 video-query pairs, and the other for paragraph captioning S_{VPC} with $|S_{\text{VPC}}|$=16533 videos. These datasets are introduced comprehensively below.

Dataset	Domain	#Videos	#Queries	MR	VPC
QVHighlights	Vlog	7100	12803	✓	✗
Charades-STA	Activity	5336	12404	✓	✓
ActivityNet Captions	Activity	10009	37421	✓	✓
Youcook2	Instruction	1188	9201	✓	✓

Table 11: Statistics of training data.

QVHighlights (Lei et al., 2021). This dataset includes 10,148 trimmed videos with an average length of 150 sec that covers daily vlogs, travel vlogs, and news events scenarios. There are in total 10,310 queries associated with 18, 367 moments. Following (Lei et al., 2021), we use *train* split for instruction tuning of moment retrieval, *test* split for evaluation, and *val* split for ablation studies. The license is *Attribution-NonCommercial-ShareAlike 4.0 International* and our usage is consistent with its license.

Charades-STA (Gao et al., 2017). The dataset contains 6,672 videos with an average duration of 30.6 sec and 16,128 moment/caption pairs. Each video is annotated with 2.4 segments on average. We use *train* split for instruction tuning and *test* for evaluation. The license is *License Non-Commercial Use* and our usage is consistent with its license.

ActivityNet Captions (Krishna et al., 2017). The dataset contains 14,934 untrimmed videos of various human activities from YouTube. On average, each video lasts 120s and is annotated with 3.7 temporally-localized sentences. The dataset is split into 10,009 and 4,925 videos for training and validation, respectively. *train* split is included in instruction tuning for both moment retrieval and video paragraph captioning. The validation set has two independent dense video captioning annotations (*val_1* and *val_2*). For moment retrieval, we evaluate on *val_2* according to prior work (Yan et al., 2023). For video paragraph captioning, we report results on the *ae-test* split following (Lei et al., 2020a; Zhou et al., 2019). For dense video captioning, we use both *val_1* and *val_2* for evaluation, by computing the average of the scores over each set for SODA_c and by using the standard evaluation tool (Krishna et al., 2017) for all other dense event captioning metrics. The license is not specified by the original authors.

YouCook2 (Zhou et al., 2018a). It has 1,790 untrimmed videos of cooking procedures. On average, each video lasts 320s and is annotated with 7.7 temporally-localized sentences. The dataset is split into 1,333 videos for training and 457 videos for validation. We use *train* split for instruction tuning and evaluate on *val* split. The license is *MIT License* and our usage is consistent with its license.

Besides, we adopt a subset of HowTo100M (Han et al., 2022) with 240K videos for automatic an-

#frames	QVHighlights			Charades-STA			ActivityNet Captions		
	R@0.5	R@0.7	mAP avg	R@0.5	R@0.7	mIoU	R@0.5	R@0.7	mIoU
25	61.5	37.4	35.0	63.4	38.0	55.0	43.6	25.9	43.9
50	65.4	47.9	41.4	67.3	46.0	58.1	46.2	28.3	46.1
75	**69.5**	**54.4**	**45.3**	**69.0**	45.6	**58.9**	48.4	**29.8**	**47.1**
100	67.8	52.8	44.7	68.4	**46.0**	58.4	**48.5**	29.3	46.5

Table 12: Effects of the number of frames on moment retrieval.

LoRA	Temporal modeling	QVHighlights			Charades-STA			ActivityNet Captions		
		R@0.5	R@0.7	mAP avg	R@0.5	R@0.7	mIoU	R@0.5	R@0.7	mIoU
✗	✗	60.6	36.4	33.2	62.3	36.8	55.0	43.1	25.9	43.8
✓	✗	66.7	49.2	39.8	67.6	44.5	58.2	44.3	27.5	45.1
✗	✓	65.5	47.0	40.4	66.8	43.4	57.3	44.9	27.7	45.3
✓	✓	**69.5**	**54.5**	**45.3**	**69.0**	**45.6**	**58.9**	**48.4**	**29.8**	**47.1**

Table 13: Effects of LoRA and temporal modeling on moment retrieval.

notation. It is a large-scale dataset of narrated videos with an emphasis on instructional videos where content creators teach complex tasks with an explicit intention of explaining the visual content on screen (Miech et al., 2019). The license is not specified by the original authors. For evaluation, we leverage three action recognition tasks: UCF101 (license not specified) (Soomro et al., 2012) , HMDB51 (CC BY 4.0) (Kuehne et al., 2011) and Kinetics-400 (CC BY 4.0) (Kay et al., 2017). Our usage is consistent with their licenses.

D Additional Results

Additional experimental results are reported in this section, including the analysis of dedicated time tokens, effects of the number of frames, effects of LoRA and temporal modeling, and the influence of different pre-training data ratios. We only run instruction tuning once and all results in Section 4 and this section are from this single model.

Extra overheads of dedicated time tokens. As mentioned in Section 4.3, new dedicated time tokens would introduce a considerably larger number of trainable parameters. In particular, given the current implementation of LLMs, it is challenging to train new tokens only without affecting the rest of parameters in the embedding layer and the final output layer. Thus, we take an alternative to tune all parameters in these two layers: given the original vocabulary size of 32000, the number of new time tokens of 75, and the hidden dimension of 4096, the total number of trainable parameters is computed as: $(32000 + 75) \times 4096 \times 2 = 262\text{M}$.

Number of frames. In addition to results presented in Table 5b, we show more complete experiments on moment retrieval and video paragraph captioning in Table 12 and Table 14. The trends are consistent with what we observed in Table 5b, where 75 frames are enough to cover all the semantic information needed for these two tasks.

#frames	YouCook2		ActivityNet	
	C	M	C	M
25	29.2	16.9	33.4	16.9
50	34.3	17.8	34.5	17.0
75	**37.8**	18.3	**34.8**	**17.3**
100	37.4	**18.5**	34.6	**17.3**

Table 14: Effects of the number of frames on video paragraph captioning.

LoRA and temporal modeling. We present more thorough and comprehensive experimental results to understand the effects of LoRA and temporal modeling. In Table 13 and 15, we can conclude that both LoRA training and temporal modeling contribute to performance gains in moment retrieval and video paragraph captioning.

LoRA	Temporal modeling	YouCook2		ActivityNet	
		C	M	C	M
✗	✗	25.7	16.9	23.0	16.0
✓	✗	32.3	17.7	34.4	17.2
✗	✓	26.5	17.4	27.6	16.8
✓	✓	**37.8**	**18.3**	**34.8**	**17.3**

Table 15: Effects of LoRA and temporal modeling on video paragraph captioning.

Base model selection. It should be emphasized that UNICORN is a generic framework to which we can flexibly utilize various LMMs as the base model with a simple re-design to take video inputs. We analyze the effects of adopting different base models like LLaVA (Liu et al., 2023b) here to justify our framework design. Specifically, we instruction-tuned LLaVA for moment retrieval and video paragraph captioning. Results are shown in Table 16 and it is expected that the performance of LLaVA variant drops compared with our InstructBLIP variant, due to the information loss from 256 frame-level tokens pooled to one token. Besides, InstructBLIP has a QFormer while LLaVA only uses a simple projection layer, which may be insufficient to align video and language. We also added an experiment with InstrutBLIP-13B and observed performance gains with a larger model size.

Base model	QVHighlights			ActivityNet	
	R@0.5	R@0.7	mAP	C	M
LLaVA-7B (Liu et al., 2023b)	66.3	51.5	42.8	33.6	16.4
InstructBLIP-7B (Dai et al., 2023)	68.2	52.3	44.8	34.6	16.9
InstructBLIP-13B (Dai et al., 2023)	**69.5**	**54.4**	**45.3**	**34.8**	**17.3**

Table 16: Comparison of different base models.

Ratios of PT dataset. We also alter the ratio of pre-training dataset and record corresponding performance in Figure 3. With only 25% of the videos, the model using UNICORN captions far outperforms other counterparts trained on all videos, demonstrating the effectiveness of our captions compared to ASR captions and its de-noised version. Meanwhile, UNICORN can generate multiple captions for the same video, with more sets of captions, we see a consistent performance gain from the model. Notably, when using 3 sets of captions ($\times$3), the performance is improved from 44.2 to 50.0 for R@0.5.

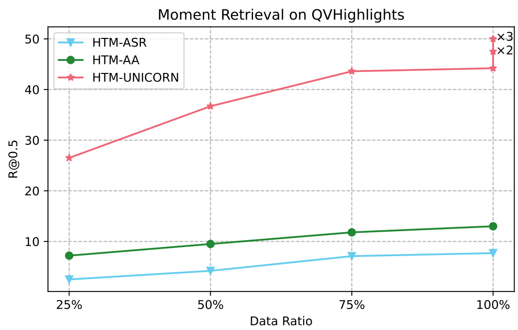


Figure 3: Zero-shot moment retrieval on QVHighlights (*val*) under different data ratios.