IPFormer: Instance Prompt-guided Transformer for Multi-modal Multi-shot Video Understanding

Yujia Liang[1], Jile Jiao[2], Xuetao Feng[2], Xinchen Liu[1], Kun Liu[1], Yuan Wang [2], Zixuan Ye[3], Hao Lu[3], Zhicheng Wang[3*],

[1]JD Explore Academy
[2]Deepeleph Intelligent Technology
[3]School of AIA, Huazhong University of Science and Technology
firstAuthor@liangyujia.3@jd.com,

Abstract

Video Large Language Models (VideoLLMs), which adopt large language models for video understanding, have been demonstrated for single-shot videos. However, they usually struggle in multi-shot videos with frequent shot changes, varying camera angles, etc., which makes VideoLLMs hardly answer questions about multiple instances or shots over the whole video. We attribute this challenge to two issues: 1) the lack of multi-shot multi-instance annotations of existing datasets, and 2) the negligence of instance-aware modeling of current VideoLLMs. Therefore, we first introduce a new dataset termed **MultiClip-Bench**, featuring dense descriptions and question-answering pairs tailored for multi-shot and multi-instance scenarios. Moreover, since the existing VideoLLMs neglect the explicit modeling of instance-related features, we propose a novel Instance Prompt-guided Transformer, named **IPFormer**, to achieve instance-aware video understanding. In the IPFormer, we design a simple but effective instance-aware feature injection module, which encodes instance features as instance prompts via an attention-based connector. By this means, IPFormer can aggregate instance-specific information across multiple shots. Extensive experiments not only show that our dataset and model significantly improve multi-shot video understanding. but also show that our MultiClip-Bench can provide valuable training data and benchmarks for various video understanding tasks.

Code — https://github.com/babadaiwo/IPFormer.git

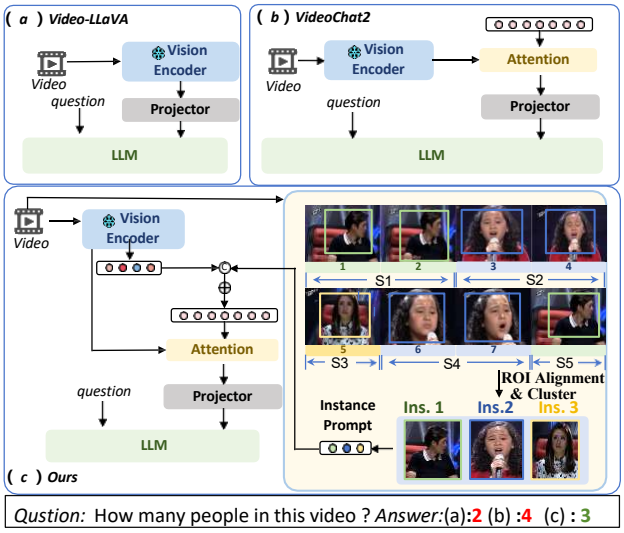

Figure 1: Existing Model Struggles with Discontinuous Instance Appearances.**(a)** uses full projection, while **(b)** reduces video tokens via attention mechanisms. Existing models fail to distinguish characters, leading to counting errors. **(c)** Our method compresses tokens and uses detectors and clustering to generate instance prompts, to guide attention fusion across scenes and enhance accuracy.

Introduction

Large Multi-modal Models (LMMs) (Alayrac et al. 2022; Driess et al. 2023; Huang et al. 2023; Li et al. 2023a) built by connecting Large Language Models (LLMs) as the brain with specialized encoders, have yielded remarkable progress in tasks involving multiple modalities. In the domain of video, recent Video Large Language Models (VideoLLMs) (Li et al. 2023c, 2024; Liu et al. 2024b; Bai et al. 2025) have customized domain-specific model architectures (Team et al. 2024; Li et al. 2023c; Qu et al. 2024) and datasets/benchmarks, significantly advancing many video understanding problems such as long video understanding (Liu et al. 2024a; Song et al. 2024), fine-grained understanding, and temporal localization (Wang et al. 2024). Although current VideoLLMs perform well in most scenarios, they exhibit notable deficiencies in multi-shot scenes, *e.g.*, video clips captured from various angles of view or different scenes. Compared to single-shot videos, multi-shot ones involve more visual clues and more complex object relationships, requiring advanced information extraction and instance-level comprehension capabilities. Fig. 1 illustrates a failure case in a multi-shot scenario. Given video clips from different cameras depicting the same talent show, the models are asked to count the number of people who have ever appeared. Surprisingly, for this seemingly simple question, Video-LLaVA (Lin et al. 2023) and VideoChat2 (Li et al. 2024) answered 2 and 4 respectively, while the ground truth is 3.

Why do existing VideoLLMs fail in such an easy case?

*Corresponding author. Zhicheng Wang

We attribute the failure to two key factors: data and model limitations. First, existing datasets lack sufficient multi-shot training data, with annotations primarily focused on single-shot videos featuring limited camera movement and scene changes, and without explicit instance references. In contrast, the descriptive quality of high-dynamic multi-shot videos remains substandard. Those multi-shot annotations tend to provide only brief overviews of the video, while lacking details and depth. Moreover, on the test end, the widely used benchmarks(Krishna et al. 2017; Maaz et al. 2024; Li et al. 2024; Fu et al. 2024) can hardly reflect the ability for multi-shot video understanding due to the absence of relevant test cases.

To fill this gap, we first introduce a multimodal video understanding benchmark called **MultiClip-Bench**. The creation of MultiClip-Bench involves both video annotation and question-answer (QA) formulation. For annotation, we segment the video into key frames, annotate each key frame with character IDs, and label the video with dense captions and character information. For QA pairs, we also focus on the multi-shot scenarios, particularly on cases where characters appear discontinuously across multiple scenes or have brief appearances, and generate a large number of QA pairs accordingly. Finally, we manually selected a subset of $2,750$ pairs to constitute the test set, with answers verified, refined, and categorized to form the benchmark.

Although significant improvement is observed when using the MultiClip training set, we found several typical failure cases of current models when handling multi-shot videos. Two significant deficiencies of current VideoLLMs are identified: i) the lack of instance consistency preservation during scene transitions; and ii) the occlusion of information about briefly appearing characters caused by large data volumes in the video. To address these challenges, we propose a new model, **IPFormer**, which utilizes *instance prompts* to guide attention for more effective instance feature preservation. Specifically, we sample and aggregate features from bounding-box regions to form instance features, and inject them as visual prompts into the query to guide attention. Besides significant performance gains, it reduces tokens to less than 10% of the baseline, cuts training time by 75%, and boosts inference speed.

Experimental results demonstrate that our proposed model performs strongly on both existing benchmarks and our MultiClip-Bench, with further performance improvements observed when utilizing our proposed training dataset. Specifically, compared to the baseline model Video-LLaVA (Lin et al. 2023), our model achieves a significant improvement of **2.3%** on MultiClip-Bench. With the support of our training dataset, the performance on MultiClip-Bench increases by **8.5%**. Noticeable performance gains are also observed on other datasets. For example, datasets such as NExT-QA (Xiao et al. 2021) and IntentQA (Li et al. 2023b) exhibit particularly remarkable enhancements. This further validates the value of our dataset in addressing the common limitations of multi-shot sences.

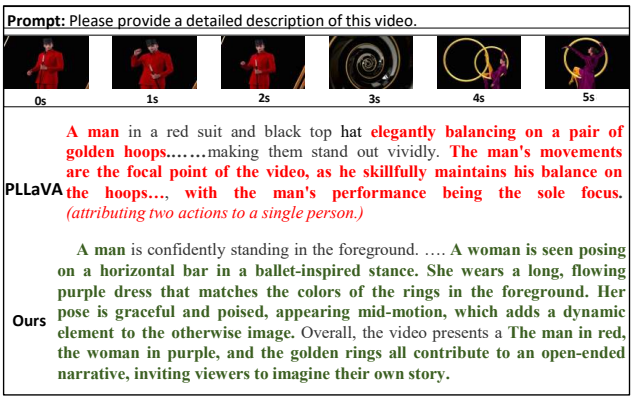

Figure 2: Challenges in Multi-shot Scenarios.

Related Work

Video Benchmarks. Video benchmark evaluations can be divided into comprehensive and task-specific sets. Modern frameworks include MMBench (Liu et al. 2024b), Video-MME (Fu et al. 2024), and MVBench (Li et al. 2024). Video-MME enhances data diversity through domain hierarchies (6 categories, 30 subclasses) and balanced video durations, while MVBench covers 20 task types for holistic evaluation. Task-specific benchmarks focus on distinct capabilities: Perception Test (Patraucean et al. 2023) for spatiotemporal reasoning, FunQA (Xie et al. 2024) for counter-intuitive scenarios, NExT-QA (Xiao et al. 2021) for object interactions, and IntentQA (Li et al. 2023b) for user intent understanding. However, most existing benchmarks focus on simple scene transitions, offering limited complexity and failing to address the challenges of multi-shot scenes, such as character consistency across discontinuous segments, short-duration segment interpretation, and soon.

Video Feature Alignment. In MLLMs, image or video tokens are flattened and projected after feature extraction, then concatenated with text before being fed into the LLM. While this approach is simple and effective, the large number of tokens—especially from videos—greatly increases LLM computational costs. As MLLMs are increasingly used in real-time applications (Song et al. 2024; Bai et al. 2025), improving inference efficiency and supporting more frames have become critical, particularly for multi-shot scenarios. The main strategy is to reduce the number of tokens input to the LLM. Token compression methods include downsampling via average pooling (Xu et al. 2024a; Bai et al. 2025), fast-slow stream merging (Xu et al. 2024b; Qu et al. 2024), memory-based fixed-length representations (Song et al. 2024), and attention-based compression with Q-former (Li et al. 2023c). However, important information can be lost during compression, especially in complex multi-shot scenes. To address this, we propose an efficient compression scheme tailored for multi-scene applications, enabling the model to process more frames while effectively preserving essential information.your conference for submission details.

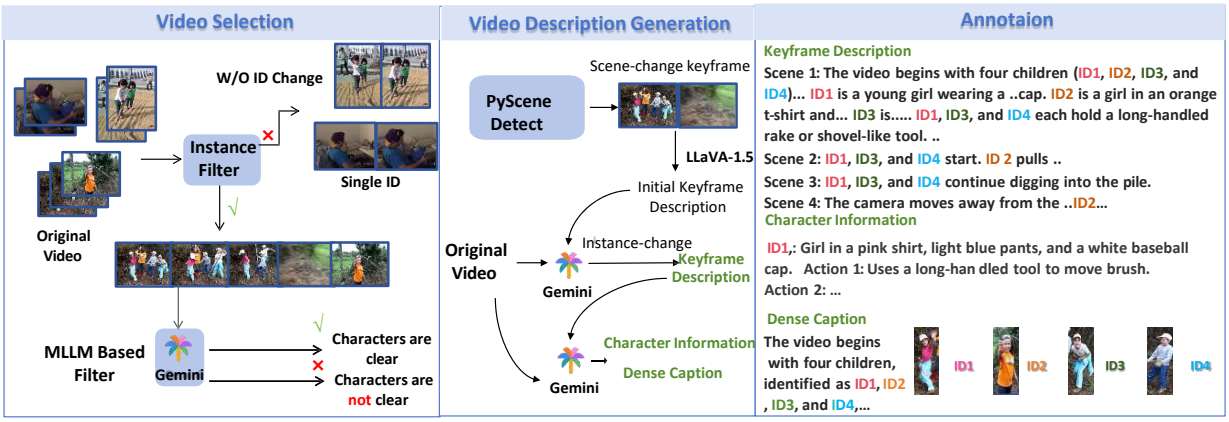

Figure 3: Description Annotations of MultiClip-Bench. First, we filter the target videos, and then use close-source large video models to generate the three type elements of video: **keyframe description**, **character information**, and **dense caption**.

Dataset	Avg-Instance	Video-Shots	Caption(C.)	Keyframe-Level C.	Istance Change Level C.	Istance Reference C.
WebVid-10M	N/A	4.5	short			
OpenVid-1M	1.2	3.7	short			
ActivityNet Captions	N/A	5.5	short			
YouCook2	1.1	1.2	short	✔		
InternVid	4.6	3.5	short	✔		
Video-ChatGPT	N/A	5.3	long	✔		
VideoGPT+	N/A	5.2	long	✔		
Ours	6.2	6.4	long	✔	✔	✔

Table 1: A comparison of the existing video caption datasets and our MultiClip-Bench.

MultiClip-Bench

Multi-Shot Description Generation

Some widely used video LLMs, such as PLLaVA (Xu et al. 2024a), claim strong video captioning capabilities. However, as shown in Figure 2, it struggles with easy multi-shot videos, often failing to recognize even simple scene transitions, largely due to the lack of tailored multi-shot video data. To address this, we introduce MultiClip-Bench, the first multi-shot video dataset with both a training set and a manually verified test set. As illustrated in Figure 3, we design an efficient data engine pipeline for video selection and description generation.

Video Selection. We aim to address instance consistency and omission in multi-shot scenarios, we propose using character changes as a proxy. Additionally, We only retain videos up to two minutes in length, focusing on multi-scene content and excluding interference from other issues such as long duration.We then employ a coarse-to-fine filtering strategy. In the first stage, a filter based on person instances is employed, using OC-SORT (Cao et al. 2023) to perform coarse filtering based on instance tracking in order to analyze the trajectories of instance IDs. Videos with ID transition exceeding 5 occurrences are retained. In the second stage, we establish two key criteria for the strong close-source video LLM Gemini-1.5-Pro-flash (Team et al. 2024) to filter videos: i) Instance ID switches in the video, which require transitions in instance identities; ii) Clear identification, limiting foreground and background characters to balance scene complexity and computational feasibility.

Video Description Generation. The next step is to obtain descriptions. However, commonly used dense descriptions are inadequate for multi-shot videos, as they often miss character associations across scenes and overlook short or less important segments. To address this, we introduce two new annotations: **keyframe descriptions** with instance IDs and **character information**, which respectively emphasizing the consistency of characters and their features and actions. To obtain additional annotations, we propose an automated framework. As shown in Figure 3, we first select keyframes using PySceneDetect and then use LLaVA-1.5 to generate detailed description for the keyframes, serving as the initial keyframe description. Then we further use Gemini to refine the initial keyframe description with the original video, utilizing its strong ability to detect instance ID changes and behavioral shifts (see Supplementary Materials for details). Finally, by combining the refined keyframe descriptions and the original video, Gemini produces both dense captions and character information. The resulting video description thus includes three key elements: keyframe descriptions, character information, and dense captions. We compared other video caption datasets, as shown in Table 1. Our dataset includes more instances (N/A for uncountable cases) and more shots per video. And compared with traditional keyframes-level captioning methods based on PySceneDetect, which

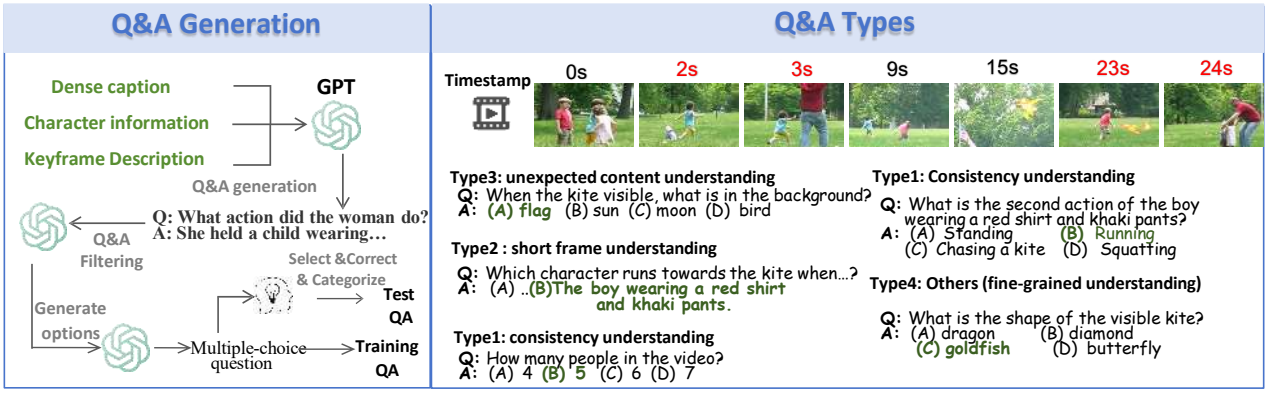

Figure 4: QA Generation Pipeline and MultiClip-Bench Examples. Questions fall into four types: consistency, short-frame, unexpected content, and others. Type 1 involves instance re-identification and discontinuous actions. Type 2 focuses on events between seconds 23-24 (short-frame). Type 3 addresses non-critical background details.

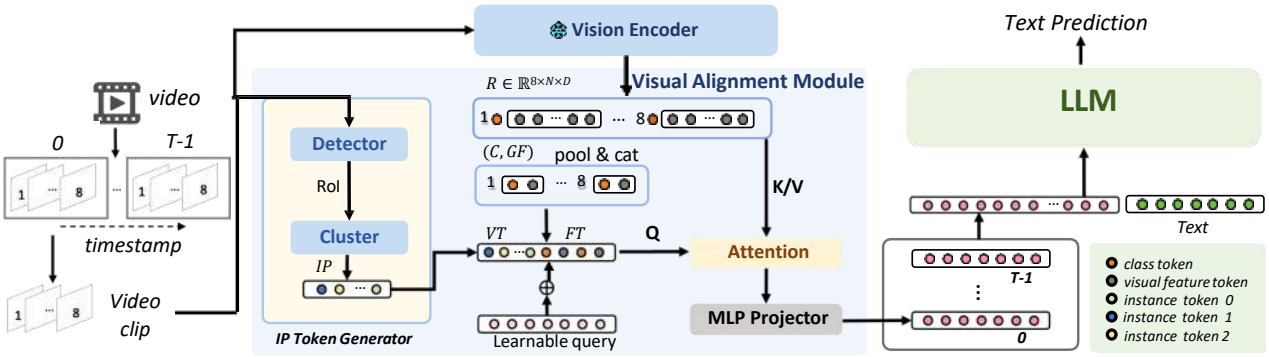

Figure 5: The pipeline of IPFormer. Video slices are sampled via a sliding window, and a video encoder extracts features from each slice. Frame-level (cls tokens and global average pooling) and instance-level (from the IP Token Generator) information are separately extracted and concatenated. This combined information is added to a learnable query, optimized via an attention mechanism, and projected by an MLP for text alignment. Finally, aligned visual features from all segments are fused with text features and input into an LLM for further processing.

are unable to capture instance changes, our approach segment clips according to instance changes and each caption has explicit instance ID.

Our video data comes from multiple sources, including Kinetics-710 (Li et al. 2022), VideoChatGPT (Maaz et al. 2023), VideoChat (Li et al. 2023c), YouCook2 , NExTQA (Xiao et al. 2021), WebVid (Yang et al. 2021), and EgoQA (Grauman et al. 2022). After filtering and annotation, each video contains the three elements as shown in Figure 3, resulting in descriptive dataset with 23k high-quality video-text pairs (7.6k videos) tailored for multi-shot scences.

QA Pairs of Multi-Shot Scenes

After obtaining the key elements for multi-shot video descriptions, we generate QA pairs for the training and test sets. These QA pairs not only enhance instruction-following but also improve comprehension. The generation process and examples are shown in Figure 4.

We first identify three main challenges in understanding multi-shot videos and generate QA pairs for each: **i) Consistency Understanding**: Focusing on identifying the same characters across non-consecutive keyframes, including tasks such as action continuity, action counting, temporal reasoning, and counting characters. **ii) Short Frame Understanding**: Addressing difficulties in capturing visual cues from brief keyframes, with questions targeting instantaneous actions, appearance details, and other visual information. **iii) Unexpected Content Understanding**: Handling non-critical content that deviates from the main theme, with questions about individuals and objects in these unexpected scenes. These categories comprehensively cover the primary challenges of multi-shot transitions.

With the proposed QA types, we employ GPT-4 to create 10 QA pairs per video, focusing on multi-instances scenes with shot transitions. Then we pass the generated initial QA pairs to GPT-4 for refinement, where 4 to 6 high-quality QA pairs are retained. Finally, three incorrect options are added to each question, converting them into a multiple-choice format with GPT-4. The videos are sourced from the collection in the previous stage. The training set contains

45.5k video-text pairs, and the test set contains 2.75k video-text pairs, with no overlap between the two sets. The manually curated, corrected, and categorized test set includes 800 consistency understanding questions, 703 unexpected content understanding questions, 562 short-frame understanding questions, and 685 other questions covering fine-grained tasks and general comprehension.

Visual-Prompt Former for Video LLM

In addition to the MultiClip-Bench, we propose IPFormer, which is adapted for multi-shot video understanding. This section details the design of the model.

Pipeline of IPFormer

Our model built on Video-LLaVA, consists of three main modules as illustrated in Figure 5: a visual encoder, a visual alignment module (incorporation the IP Token Generator, attention mechanism and MLP), and a large language model (LLM). Due to the fixed number of frame constraint for video encoders, we employ a sliding-window sampling approach. Specifically, for an input video, F frames are uniformly extracted. Using a sliding window of 8 frames, we perform non-overlapping sampling, resulting in $T = \lfloor \frac{F}{8} \rfloor$ slices. Each slice is processed separately by the visual encoder and the visual alignment module. The processed slices are then concatenated and fed into the LLM along with the text. We innovatively introduce the visual alignment module, where instance feature tokens generated by the IP Token Generator are used as prompts to guide the attention mechanism in aggregating instance features.

Vision Alignment Module of IPFormer

Motivation: Based on our research on data, we assume that visual alignment modules for multi-shot video understanding should meet two requirements: i) effective compression of vision tokens to handle numerous frames, and ii) instance awareness to capture critical instance information across scenes. However, current visual alignment modules, whether full projection or compression before projection, encode instance features in a discrete or lossy manner, inevitably losing instance-level information. To address this, we propose an instance-based compressor as a visual alignment module, which reduces key feature loss while compressing visual tokens. **Method:** To enable the attention mechanism to capture more critical information, we introduce "anchors" (Ren et al. 2015; Meng et al. 2021) when initializing learnable query tokens, inspired by Conditional-DETR (Meng et al. 2021). These anchors guide the learnable tokens to extract relevant features from frames, thereby preserving key information. Thus, we input both instance-level and frame-level information as "anchors" into the learnable query tokens. Specifically, prior to encoding, the video is pre-processed into images of size 224×224 for 8 frames. The video encoder then processes these 8 frames into $\mathbf{R} \in R^{8\times N\times D}$, where N is the number of tokens per frame, and D is the vector dimension. Here, $N = 256+1$, where 256 denotes the number of image tokens per frame, called $\mathbf{N}_I \in R^{256\times D}$, and 1 is the class token per frame, referred to as $\mathbf{C} \in R^{1\times D}$. Next, global average pooling in performed on $\mathbf{N}_I$ to obtain the global feature of each frame, $\mathbf{GF} \in R^{1\times D}$. We use $\mathbf{GF}$ and $\mathbf{C}$ as global information for each frame (frame-level guidance). To enhance frame information learning, $\mathbf{GF}$ and $\mathbf{C}$ are repeated each X times (X=5), then concatenated to obtain the frame token for each frame: $\mathbf{FT} = (\mathbf{C} \times X, \mathbf{GF} \times X) \in R^{(X\times 2)\times D}$. In addition, we also obtain instance features through the IP Token Generator. The instance prompt tokens for this segment are represented as $\mathbf{IP} \in R^{V\times D}$, where V is the number of instance tokens. We concatenate $\mathbf{FT}$ of 8 frames and $\mathbf{IP}$, resulting in the visual tokens for 8 frames as $\mathbf{VT} \in R^{8\times(X\times 2)\times D+V\times D}$. Finally, $\mathbf{VT}$ is added to the learnable queries to optimize them, guiding the queries to aggregate video features through the cross-attention mechanism and better capture relevant instance or other useful information. The attention output is then projected by an MLP to align with the textual space.

IP Token Generator. The IP Token Generator generates instance tokens as prompts. Specifically, we employ a category-agnostic detector (Zhu et al. 2020) initialized with parameters from (Ma et al. 2024). This yields a set of bounding boxes, which are refined using NMS (Neubeck and Van Gool 2006). For each frame, we retain only M (fewer than 10) candidate boxes, truncating excess and padding with zeros as needed. These candidate boxes are processed using global RoI pooling on the visual feature maps of each frame to obtain instance features. Next, we compute the cosine similarity between the instance features across all frames within a video slice. Using a threshold of 0.9, we adopt an iterative grouping strategy: starting from the first ungrouped instance, we collect all remaining instances with similarity above the threshold to form a group. This process repeats until no further groupings can be made. Any remaining instances form independent groups. For each group, we compute the channel-wise average to obtain aggregated instance features, termed Instance Prompts. Each slice contains up to V (80) instance tokens, with zero-padding if fewer tokens are present. Clustering balances the number of instances, preventing frequent instances from overwhelming sparse ones. which is beneficial for short frame.

Experiments

Implementation Details

Model Details: To validate the effectiveness of the proposed modules, we use Video-LLaVA (Lin et al. 2023) as the baseline, maintaining the same training settings to ensure the only difference are in the model modifications. Specifically, visual encoder from LanguageBind (Zhu et al. 2023) is used, initialized from OpenCLIP-L/14 (Radford et al. 2021), with a resolution of 224×224. For large language model, we use the LLM in LanguageBind initialized from Vicuna-7B v1.5 (Zheng et al. 2023). The text tokenizer is from LLaMA with about 32,000 classes.

Regarding our design, in addition to the two-layer MLP full projection with GeLU (Hendrycks and Gimpel 2016) for the visual compressor, we adopt the approach of BLIP2 (Li et al. 2023a) and VideoChat2 (Li et al. 2024), implementing an attention parameter compression module using pre-

Method	LLM size	Frames	Tokens	Consistency uds	Short uds	Unexpected content	Others	Mean
Video-LLaVA (Lin et al. 2023)	7B	8	2056	32.9	40.0	46.2	60.7	44.5
VideoChat2 (Li et al. 2024)	7B	16	96	33.4	32.3	47.6	58.1	42.9
ST-LLaVA (Qu et al. 2024)	7B	100	2304	31.1	38.8	48.3	56.8	43.5
PLLaVA (Xu et al. 2024a)	7B	16	2304	36.5	43.0	55.9	63.2	49.4
LLaVA-MINI (Zhang et al. 2025)	8B	64	64	26.8	40.9	53.2	58.6	44.9
Ours (w/o MultiClip)	7B	16	320	37.7	43.1	48.4	61.9	48.1
Ours	7B	16	320	46.1	53.6	60.4	66.3	57.0
$\Delta Acc.$	-	-	-	**+8.4**	**+10.5**	**+12.0**	**+4.4**	**+8.9**
Ours	7B	48	960	**48.6**	**56.7**	**62.5**	68.5	**58.8**
GPT-4o-mini (Abacha et al. 2024)	8B	16	Unk	44.7	44.6	55.3	59.1	50.9
GPT-4V(Yang et al. 2023)	Unk	8	Unk	34.6	31.5	45.3	52.4	41.8
Qwen2.5-vl (Bai et al. 2025)	72B	16	Unk	43.5	**58.3**	**62.4**	72.8	58.5
Gemini-1.5-pro (Team et al. 2024)	Unk	16	Unk	**53.4**	52.5	62.0	**74.4**	**60.5**

Table 2: Comparison with other models on MultiClip-Bench. Our model demonstrates superior performance in multi-shot scenarios compared to other methods. Best performance is **boldface**. The highlight in blue is baseline.

Method	Frame	Token	Train Time	FPS	Act-Net
Video-LLaVA	8	2056	41h	1.1	45.3
Ours	8	160	10h	3.3	45.5
Ours	16	320	-	2.0	46.1

Table 3: Efficiency comparison between Video-LLaVA (baseline) and ours. Our model outperforms the baseline in both efficiency and performance.

Method	MSVD	MSRVTT	ActivityNet
FrozenBiLM (Yang et al. 2022)	33.8	16.7	25.9
Video-LLaMA (Zhang, Li, and Bing 2023)	51.6	29.6	12.4
LLama-Adapter (Zhang et al. 2023)	54.9	43.8	34.2
Video-ChatGPT (Maaz et al. 2023)	64.9	49.3	35.3
VideoChat (Li et al. 2023c)	56.3	45.0	26.5
Video-LLaVA (Lin et al. 2023)	70.7	59.2	45.3
Chat-UniVL (Jin et al. 2024)	65.5	54.6	45.8
MovieChat (Song et al. 2024)	75.2	52.7	45.7
VideoChat2 (Liu et al. 2024b)	70.0	54.1	49.1
Vista-LLama (Ma et al. 2023)	65.3	60.5	48.3
LLAMA-VID(Li, Wang, and Jia 2024)	70.0	58.9	47.5
ST-LLM (Liu et al. 2024a)	**74.6**	**63.2**	**50.9**
Ours	73.8	**63.2**	50.1

Table 4: Quantitative comparisons on common open-ended video question answering.

Method	**Video-MME**	**MVBench**
Video-LLaVA(Lin et al. 2023)	39.9	42.2
PLLaVA (Xu et al. 2024a)	43.1	46.5
Video-LLaMA (Zhang, Li, and Bing 2023)	-	34.7
LLaMA-Adapter (Zhang et al. 2023)	-	31.7
Video-ChatGPT (Maaz et al. 2023)	-	32.7
VideoChat(Li et al. 2023c)	39.2	35.5
VideoChat2 (Li et al. 2024)	43.8	51.1
LLaVA-MINI (Zhang et al. 2025)	41.4	44.5
Ours	42.8	48.3

Table 5: Quantitative comparison on general Multiple-choice question-answering.

Method	NExT-QA	Egoschema	IntentQA
Video-LLaVA (Lin et al. 2023)	60.5	37.0	-
Video-LLaMA2 (Cheng et al. 2024)	-	51.7	-
MovieChat+ (Song et al. 2024)	54.8	56.4	-
Vista-LLaMA (Ma et al. 2023)	60.7	-	-
DeepStack-L (Meng et al. 2025)	61.0	38.4	-
M^3 (Cai et al. 2024)	63.1	36.8	58.8
IG-VLM (Kim et al. 2024)	63.1	35.5	60.3
SF-LLaVA (Xu et al. 2024b)	62.4	47.6	60.1
TS-LLaVA (Qu et al. 2024)	66.5	**50.2**	61.7
ours	**70.6**	50.0	**75.3**

Table 6: Quantitative comparison on Multiple-choice question-answering in specific fields.

trained BERTbase. For the external detector, we adopt the Region Proposer parameters from Groma (Ma et al. 2024), implementing a modified binary classifier based on Deformable DETR (Zhu et al. 2020).

Training Details: Our training process follows Video-LLaVA (Lin et al. 2023) and is divided into two stages: modality alignment pretraining and instruction fine-tuning.

During *modality alignment pretraining*, we follow Video-LLaVA using 558K image-text pairs from LAION-CC-SBU (Liu et al. 2023) and 702K video-text pairs from Valley (Luo et al. 2023). The visual alignment module is trained for one epoch (batch size 256) with AdamW(lr=$1e^{-3}$, warmup=0.03), freezing other parameters. Each video contains 8 frames at a 224×224 resolution.

For the *instruction tuning* stage, we use the same base data as Video-LLaVA (665K image-text pairs from LLaVA-v1.5 and 100K video-text pairs from Video-ChatGPT). To further enhance performance, we additionally selected 353K instruction pairs from VideoChat2 (Li et al. 2024), and 68K video-text pairs from our MultiClip dataset. We also increase the number of sampled frames from 8 to 16 using a sliding window approach to benefit multi-shot scenes. The training settings remain consistent with Video-LLaVA: keeping video encoders frozen while fine-tuning LLM parameters with a batch size of 128.

Efficiency of IPFormer

To address the computational bottleneck in Video-LLaVA's visual processing, we propose replacing its fully connected layers with an attention-based visual compressor. We validate its efficiency through comprehensive timing experiments, as shown in Table 3. The results demonstrate that this modification significantly reduces visual tokens while maintaining model performance. In subsequent experiments, unless otherwise noted, inference is performed with 16 frames.

Result on MultiClip-Bench

We compare our model with existing open-source video multimodal models on our self-constructed MultiClip-Bench dataset, with results shown in Table 2. By comparing Video-LLaVA and Ours (w/o MultiClip), it is obvious that our model design leads to a significant improvement (**+3.6%**) When incorporating our proposed training dataset, performance improves further by **+8.9%**. Additionally, increasing the frame number can help understanding.

We also test several state-of-the-art large multimodal models (most of them are closed-source) using a multi-image evaluation approach. Among these, Qwen2.5-VL (Bai et al. 2025) and Gemini-1.5-pro (Team et al. 2024) each demonstrate their strengths in different categories.

Results on other video benchmarks

Given the significant differences in current training data and methodologies, To ensure fairness, **we use Video-LLaVA as the baseline model under the same settings.** We primarily compare the performance of our proposed model with the baseline and other models trained under similar conditions, thereby objectively and fairly demonstrating the advantages of our model and dataset.

Besides the multi-shot scenes, we also evaluate our model on various video benchmarks to demonstrate its versatility. For open-ended Video Question Answering tasks, including MSVD-QA (Chen and Dolan 2011), MSRVTT-QA (Xu et al. 2016), and ActivityNet-QA, where answers are typically short phrases, we follow Video-LLaVA to use GPT-3.5-turbo for accuracy evaluation, with results shown in Table 4. We also assess performance on more general Multiple-choice question-answering in Table 5, like MVBench, Video-MME, our model demonstrates superior performance among visual compression models. Furthermore, we validate on more challenging multi-choice benchmarks in Table 6. Through combined improvements in data and model architecture, our approach significantly outperforms other methods. This performance can be attributed to the effective coverage of multi-shot scenes and instance-centric questions within our dataset.

Ablation study

	Frame Query	Instance Query		ActivityNet	MultiClip
		no-cluster	cluster		
S1				40.9	42.3
S2	✔			42.6	43.1
S3	✔	✔		43.6	44.6
S4	✔		✔	45.5	46.2

Table 7: Design of visual compression module.

Design of the Visual Compression Module: Here we verify the effect of each design aspect within our visual compression module, with results shown in Table 7. Comparison of S1 and S2 shows that incorporation of frame queries leads to 1.7% performance gain compared to the original learnable queries in Q-Former. Additionally, injecting instance to queries can further improve the performance, but without clustering, the gain is limited. With clustering of instance tokens, performance increases to 46.2 on MultiClip.

Visual Alignment Module		Metric			
structure	Token	Activity	MSRVTT	MVBench	MultiClip
Full-projection	8*256	45.3	60.2	42.2	44.5
Avg-pooling (2*2)	8*64	45.0	59.7	41.7	43.8
QFormer (S1)	160	40.9	56.9	40.1	42.3
HCA Clustering	-	44.8	58.5	41.8	43.5
ours (S4)	8*10+80	45.5	61.8	42.3	46.2

Table 8: Comparison of different visual alignment methods.

	Data source	ActivityNet	MVBench	MuitiClip
D1	Video-LLaVA	45.8	43.7	47.1
D2	+VideoChat2(353k)	48.5	47.5	48.1
D3	+ MultiClip(68k)	50.1	48.3	57.0

Table 9: The effect of training dataset on our model.

Comparison of Different Visual Alignment Methods: Under identical experimental settings, we compare full projection, some commonly compression methods, and our IP-Former in Table 8. Full projection, commonly used in Video-LLaVA, does not compress the tokens and thus minimizes information loss. Average pooling has a low compression rate, resulting in only marginal performance drops. In contrast, the original Q-Former (S1 in Table 7) results in significant information loss, which largely impacts the performance. However, with our query design (S4 in Table 7), the compression architecture can surpass the full-projection.

	Data source	ActivityNet	MVBench	MuitiClip
D1	Original Dataset	56.5	46.4	49.4
D2	+MultiClip(68k)	57.4	47.6	57.6

Table 10: Generalization Capability of MultiClip.

Effect of MultiClip: To ensure a completely fair evaluation of our MultiClip, as shown in Table 9, adding VideoChat2 to the base data of Video-LLaVA improves performance on general datasets. Furthermore, using only 68k MultiClip training samples, our method achieves significant gains on the multi-shot MultiClip dataset, while also improving results on other general datasets. To further validate the effectiveness of our proposed MultiClip, we also incorporate it into PLLaVA (Xu et al. 2024a) for evaluation. The performerce is shown in Table 10.

Conclusion

Recognizing a gap not previously addressed in the handling of multi-shot videos with VideoLLMs, we identify the deficiencies and their underlying causes. This lead us to propose proprietary datasets and a novel design. By developing an automated video annotation pipeline, we tackle data scarcity in multi-shot scenarios. Furthermore, we introduce an instance-based visual compression module for multi-scene challenges. Our dataset and model significantly enhance multi-scene video understanding and demonstrate unique advantages across various video benchmarks.

Acknowledgments

This work is supported by National Key Research and Development Program of China (NO. 2024YFE0203200).

References

Abacha, A. B.; Yim, W.-w.; Fu, Y.; Sun, Z.; Yetisgen, M.; Xia, F.; and Lin, T. 2024. Medec: A benchmark for medical error detection and correction in clinical notes. *arXiv preprint arXiv:2412.19260*.

Alayrac, J.-B.; Donahue, J.; Luc, P.; Miech, A.; Barr, I.; Hasson, Y.; Lenc, K.; Mensch, A.; Millican, K.; Reynolds, M.; et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35: 23716–23736.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-VL Technical Report. *arXiv preprint arXiv:2502.13923*.

Cai, M.; Yang, J.; Gao, J.; and Lee, Y. J. 2024. Matryoshka multimodal models. In *Workshop on Video-Language Models@ NeurIPS 2024*.

Cao, J.; Pang, J.; Weng, X.; Khirodkar, R.; and Kitani, K. 2023. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Chen, D.; and Dolan, W. B. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, 190–200.

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 preprint arXiv:2406.07476*.

Driess, D.; Xia, F.; Sajjadi, M. S.; Lynch, C.; Chowdhery, A.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; Huang, W.; et al. 2023. Palm-e: An embodied multimodal language model.

Fu, C.; Dai, Y.; Luo, Y.; Li, L.; Ren, S.; Zhang, R.; Wang, Z.; Zhou, C.; Shen, Y.; Zhang, M.; et al. 2024. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

Grauman, K.; Westbury, A.; Byrne, E.; Chavis, Z.; Furnari, A.; Girdhar, R.; Hamburger, J.; Jiang, H.; Liu, M.; Liu, X.; et al. 2022. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 18995–19012.

Hendrycks, D.; and Gimpel, K. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Huang, S.; Dong, L.; Wang, W.; Hao, Y.; Singhal, S.; Ma, S.; Lv, T.; Cui, L.; Mohammed, O. K.; Patra, B.; et al. 2023. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems*, 36: 72096–72109.

Jin, P.; Takanobu, R.; Zhang, W.; Cao, X.; and Yuan, L. 2024. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13700–13710.

Kim, W.; Choi, C.; Lee, W.; and Rhee, W. 2024. An image grid can be worth a video: Zero-shot video question answering using a vlm. *IEEE Access*.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Carlos Niebles, J. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, 706–715.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 19730–19742.

Li, J.; Wei, P.; Han, W.; and Fan, L. 2023b. Intentqa: Context-aware video intent reasoning. In *Proceedings of the IEEE/CVF international conference on computer vision*, 11963–11974.

Li, K.; He, Y.; Wang, Y.; Li, Y.; Wang, W.; Luo, P.; Wang, Y.; Wang, L.; and Qiao, Y. 2023c. Videochat: Chat-centric video understanding. *arXiv preprint 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22195–22206.

Li, K.; Wang, Y.; He, Y.; Li, Y.; Wang, Y.; Wang, L.; and Qiao, Y. 2022. Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. *arXiv preprint arXiv:2211.09552*.

Li, Y.; Wang, C.; and Jia, J. 2024. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, 323–340.

Lin, B.; Ye, Y.; Zhu, B.; Cui, J.; Ning, M.; Jin, P.; and Yuan, L. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36: 34892–34916.

Liu, R.; Li, C.; Tang, H.; Ge, Y.; Shan, Y.; and Li, G. 2024a. St-llm: Large language models are effective temporal learners. In *European Conference on Computer Vision*, 1–18.

Liu, Y.; Duan, H.; Zhang, Y.; Li, B.; Zhang, S.; Zhao, W.; Yuan, Y.; Wang, J.; He, C.; Liu, Z.; et al. 2024b. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, 216–233.

Luo, R.; Zhao, Z.; Yang, M.; Dong, J.; Li, D.; Lu, P.; Wang, T.; Hu, L.; Qiu, M.; and Wei, Z. 2023. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*.

Ma, C.; Jiang, Y.; Wu, J.; Yuan, Z.; and Qi, X. 2024. Groma: Localized visual tokenization for grounding multimodal large language models. In *European Conference on Computer Vision*, 417–435.

Ma, F.; Jin, X.; Wang, H.; Xian, Y.; Feng, J.; and Yang, Y. 2023. Vista-llama: Reliable video narrator via equal distance to visual tokens. *arXiv preprint arXiv:2312.08870*.

Maaz, M.; Rasheed, H.; Khan, S.; and Khan, F. 2024. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arXiv preprint arXiv:2406.09418*.

Maaz, M.; Rasheed, H.; Khan, S.; and Khan, F. S. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Meng, D.; Chen, X.; Fan, Z.; Zeng, G.; Li, H.; Yuan, Y.; Sun, L.; and Wang, J. 2021. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3651–3660.

Meng, L.; Yang, J.; Tian, R.; Dai, X.; Wu, Z.; Gao, J.; and Jiang, Y.-G. 2025. Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms. *Advances in Neural Information Processing Systems*, 37: 23464–23487.

Neubeck, A.; and Van Gool, L. 2006. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)*, volume 3, 850–855.

Patraucean, V.; Smaira, L.; Gupta, A.; Recasens, A.; Markeeva, L.; Banarse, D.; Koppula, S.; Malinowski, M.; Yang, Y.; Doersch, C.; et al. 2023. Perception test: A diagnostic benchmark for multimodal video models. *Advances in Neural Information Processing Systems*, 36: 42748–42761.

Qu, T.; Li, M.; Tuytelaars, T.; and Moens, M.-F. 2024. TS-LLaVA: Constructing Visual Tokens through Thumbnail-and-Sampling for Training-Free Video Large Language Models. *arXiv preprint arXiv:2411.11066*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Song, E.; Chai, W.; Wang, G.; Zhang, Y.; Zhou, H.; Wu, F.; Chi, H.; Guo, X.; Ye, T.; Zhang, Y.; et al. 2024. Moviechat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18221–18232.

Team, G.; Georgiev, P.; Lei, V. I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Wang, H.; Xu, Z.; Cheng, Y.; Diao, S.; Zhou, Y.; Cao, Y.; Wang, Q.; Ge, W.; and Huang, L. 2024. Grounded-videollm: Sharpening fine-grained temporal grounding in video large language models. *arXiv preprint arXiv:2410.03290*.

Xiao, J.; Shang, X.; Yao, A.; and Chua, T.-S. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9777–9786.

Xie, B.; Zhang, S.; Zhou, Z.; Li, B.; Zhang, Y.; Hessel, J.; Yang, J.; and Liu, Z. 2024. Funqa: Towards surprising video comprehension. In *European Conference on Computer Vision*, 39–57.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5288–5296.

Xu, L.; Zhao, Y.; Zhou, D.; Lin, Z.; Ng, S. K.; and Feng, J. 2024a. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*.

Xu, M.; Gao, M.; Gan, Z.; Chen, H.-Y.; Lai, Z.; Gang, H.; Kang, K.; and Dehghan, A. 2024b. Slowfast-llava: A strong training-free baseline for video large language models. *arXiv preprint arXiv:2407.15841*.

Yang, A.; Miech, A.; Sivic, J.; Laptev, I.; and Schmid, C. 2021. Just ask: Learning to answer questions from millions of narrated videos. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1686–1697.

Yang, A.; Miech, A.; Sivic, J.; Laptev, I.; and Schmid, C. 2022. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 35: 124–141.

Yang, Z.; Li, L.; Lin, K.; Wang, J.; Lin, C.-C.; Liu, Z.; and Wang, L. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1): 1.

Zhang, H.; Li, X.; and Bing, L. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.

Zhang, R.; Han, J.; Liu, C.; Gao, P.; Zhou, A.; Hu, X.; Yan, S.; Lu, P.; Li, H.; and Qiao, Y. 2023. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Zhang, S.; Fang, Q.; Yang, Z.; and Feng, Y. 2025. LLaVA-Mini: Efficient Image and Video Large Multimodal Models with One Vision Token. *arXiv preprint arXiv:2501.03895*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623.

Zhu, B.; Lin, B.; Ning, M.; Yan, Y.; Cui, J.; Wang, H.; Pang, Y.; Jiang, W.; Zhang, J.; Li, Z.; et al. 2023. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*.

Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; and Dai, J. 2020. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*.