

# Video-Panda: Parameter-efficient Alignment for Encoder-free Video-Language Models

Jinhui Yi<sup>\*1</sup> Syed Talal Wasim<sup>\*1,2</sup> Yanan Luo<sup>\*1</sup> Muzammal Naseer<sup>3</sup> Juergen Gall<sup>1,2</sup>

<sup>1</sup>University of Bonn <sup>2</sup>Lamarr Institute for Machine Learning and Artificial Intelligence

<sup>3</sup>Khalifa University

<sup>\*</sup> Equal contribution

## Abstract

We present an efficient encoder-free approach for video-language understanding that achieves competitive performance while significantly reducing computational overhead. Current video-language models typically rely on heavyweight image encoders (300M-1.1B parameters) or video encoders (1B-1.4B parameters), creating a substantial computational burden when processing multi-frame videos. Our method introduces a novel Spatio-Temporal Alignment Block (STAB) that directly processes video inputs without requiring pre-trained encoders while using only 45M parameters for visual processing - at least a 6.5× reduction compared to traditional approaches. The STAB architecture combines Local Spatio-Temporal Encoding for fine-grained feature extraction, efficient spatial downsampling through learned attention and separate mechanisms for modeling frame-level and video-level relationships. Our model achieves comparable or superior performance to encoder-based approaches for open-ended video question answering on standard benchmarks. The fine-grained video question-answering evaluation demonstrates our model's effectiveness, outperforming the encoder-based approaches Video-ChatGPT and Video-LLaVA in key aspects like correctness and temporal understanding. Extensive ablation studies validate our architectural choices and demonstrate the effectiveness of our spatio-temporal modeling approach while achieving 3-4× faster processing speeds than previous methods. Code is available at <https://github.com/jh-yi/Video-Panda>.

## 1. Introduction

Recently, large language models (LLMs) have demonstrated remarkable capabilities in understanding and generating text, catalyzing significant advancements in vision-language modeling [1, 23, 35, 44]. While initial efforts focused on image understanding, recent works have extended

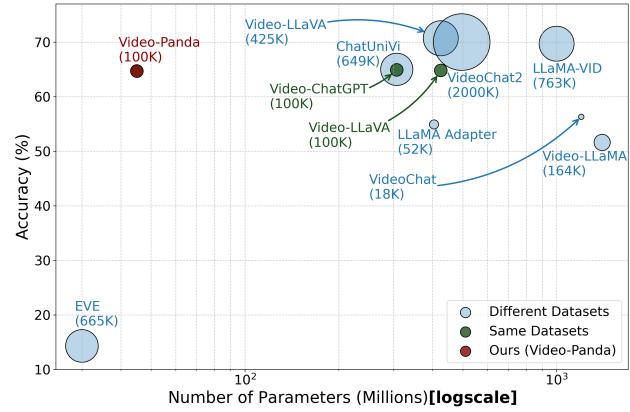


Figure 1. Model performance on MSVD-QA versus the model size of the visual component in logarithmic scale. The bubble size indicates the amount of finetuning data (in thousands). Models using the same training dataset as ours (100K samples) are shown in dark green, while those using different datasets are in blue.

these capabilities to video comprehension, enabling more complex spatio-temporal reasoning [18, 26, 42]. However, current video-language models face two major challenges.

First, existing video-language models typically rely on either heavyweight image encoders (300M to 1.1B parameters) [15, 22, 26, 43] or even larger video encoders (1B to 1.4B parameters) [18, 37, 42]. Some approaches even combine both types of encoders for enhanced feature extraction [27]. These large encoders create substantial computational overhead, particularly when processing videos with multiple frames that require repeated passes through the encoder. Furthermore, the alignment between these encoders and language models is typically achieved through either simple linear projections [43], which ignore spatio-temporal relations, or more complex mechanisms that combine linear projections with additional resampling modules such as query-transformers [18, 42], adding further computational complexity.

Second, current approaches often struggle with the inherent complexity of video understanding. Recent work has shown that naive adaptations of image-language architectures to video comprehension result in significant performance degradation when trained solely on video data [18]. This suggests that effective video-language models require specialized architectures that can capture the unique spatio-temporal relationships present in video content, rather than treating videos as simple sequences of images.

Given these considerations, we ask: *Is it possible to develop an efficient video-language model that bypasses the need for heavyweight encoders while maintaining state-of-the-art accuracy?*. To address this question, we introduce a novel encoder-free video-language model that achieves remarkable performance with significantly reduced computational overhead. The key to our approach is a specialized spatio-temporal alignment block that effectively bridges visual and linguistic information without requiring image or video encoders.

As shown in Figure 1, our approach achieves strong performance on MSVD-QA [9] video-language benchmarks but requires only a fraction of the parameters typically dedicated to visual encoding in traditional approaches. Our key contributions include:

- A novel encoder-free video-language model that achieves strong open-ended video question answering performance while using only 45M parameters for visual processing - a  $6.5\times$  reduction compared to Video-ChatGPT (307M) and  $9\times$  reduction compared to Video-LLaVA (425M).
- Strong performance on fine-grained video question answering metrics, outperforming encoder-based approaches on key aspects like correctness and temporal understanding.
- A specialized spatio-temporal alignment block that processes videos  $3\text{-}4\times$  faster than the encoder-based approaches Video-ChatGPT or Video-LLaVA while maintaining competitive accuracy.

## 2. Related Work

### 2.1. Large Language Models

The landscape of Large Language Models (LLMs) [1, 6, 7, 31, 35, 36, 44] has evolved significantly with the introduction of various commercial and open-source models. Following the release of ChatGPT [29], the AI community responded with open-source alternatives like LLaMA [35], Vicuna [10], Alpaca [33], and LLaMA 2 [36]. These models underwent instruction tuning to facilitate human-AI conversations. Notable developments include Instruct-GPT [31], which was trained based on GPT-3 [6] with 175 billion parameters through human preference alignment. While these models demonstrated powerful reasoning capa-

bilities, they were initially limited to text-only interactions, prompting research into multimodal extensions.

### 2.2. Image-Language Models

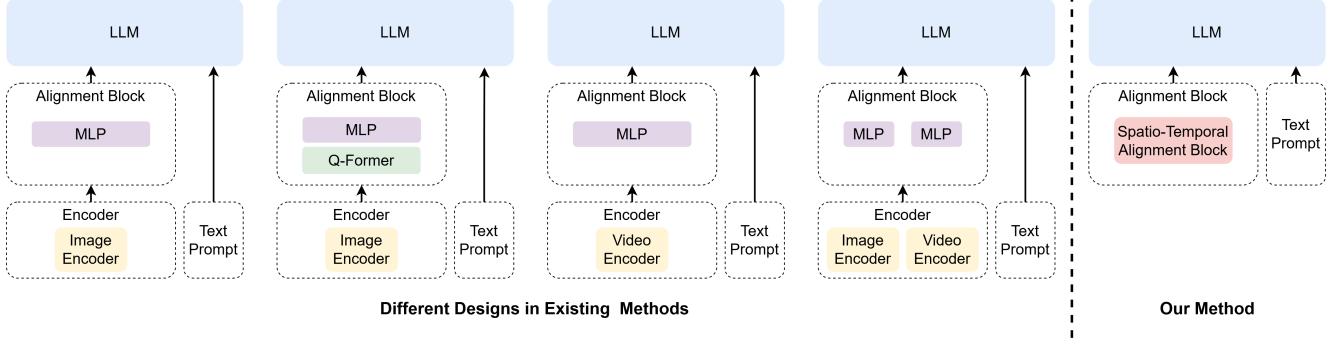
The integration of Large Language Models (LLMs) with Large Vision Models (LVMs) has produced two distinct approaches to building Image-Language Models. The first and more established approach uses encoder-based architectures, as seen in commercial offerings like Claude-3V [3], Grok-1.5V [39], MM1 [28], Gemini series [34], and GPT-4V [30]. In the open-source domain, models like BLIP-2 [16], MiniGPT-4 [46], and LLaVA [23, 24] employed learnable modules to project image features into the language space.

More recently, an encoder-free approach has emerged, pioneered by Fuyu-8B [5], which processes image inputs directly through a decoder-only network. This architecture naturally handles high-resolution images with arbitrary aspect ratios, avoiding the constraints of traditional encoder-based systems. Recent advances in this direction include EVE [11], which introduces a novel training recipe focusing on unified decoder representation and enhanced visual recognition capabilities. These encoder-free models effectively address several key challenges: resolution constraints, deployment overhead, and capacity matching between vision and language components. Using minimal public training data, models like EVE [11] have demonstrated competitive performance with encoder-based approaches while requiring fewer computational resources.

### 2.3. Video-Language Models

When extending language models to video understanding, several approaches have been proposed to handle the additional temporal dimension. Early video-capable models like Video-ChatGPT [26] introduced temporal encoding components but faced challenges regarding computational cost and limitations of the context window.

VideoChat [18] initiated an end-to-end approach by integrating video foundation models with LLMs through a learnable neural interface. This system excelled in spatiotemporal reasoning and event localization, demonstrating the potential of chat-centric video understanding. Building upon this, VideoChat2 [20] further advanced temporal understanding capabilities through progressive multi-modal training with diverse instruction-tuning data. LLaMA-Adapter V2 [13] proposed an efficient approach by unlocking more learnable parameters and introducing an early fusion strategy for visual tokens. Their joint training paradigm effectively balanced image-text alignment and instruction following tasks. Similarly, LLaMA-VID [21] used a dual-token strategy - using context and content tokens for each frame, enabling the processing of hour-long videos while preserving critical information.



**Figure 2. Existing video-language model architectures:** From left to right: Early approaches use image encoders for both image and video inputs. The alignment module aligns the embeddings of the visual modality with the language modality. The integration of Q-Former improved this alignment. Instead of a single encoder, dual encoder approaches have separate encoders for images and videos where the alignment block consists of projection layers. The additional encoders, however, make these models very heavy where the alignment module and encoders have at least 300M and sometimes over 1B parameters. In contrast, our encoder-free design (rightmost) directly processes video inputs through a novel spatio-temporal alignment block (STAB). It eliminates the need for heavyweight pretrained encoders and requires less than 50M parameters.

Video-LLaMA [42] addressed multimodal integration by incorporating both visual and auditory content through a Video Q-former for temporal understanding and an Audio Q-former built on ImageBind for audio processing. Chat-UniVi [15] took a different approach by utilizing dynamic visual tokens to create a unified representation for both images and videos, employing a multi-scale representation to capture both high-level concepts and low-level details. Video-LLaVA [22] introduced a unified approach that aligns visual representations before projecting them into the language space. Combined with joint training on mixed image and video datasets, this method demonstrated strong performance across modalities. *To the best of our knowledge, an encoder-free video-language model does not exist, representing a notable gap in the literature.*

### 3. Method

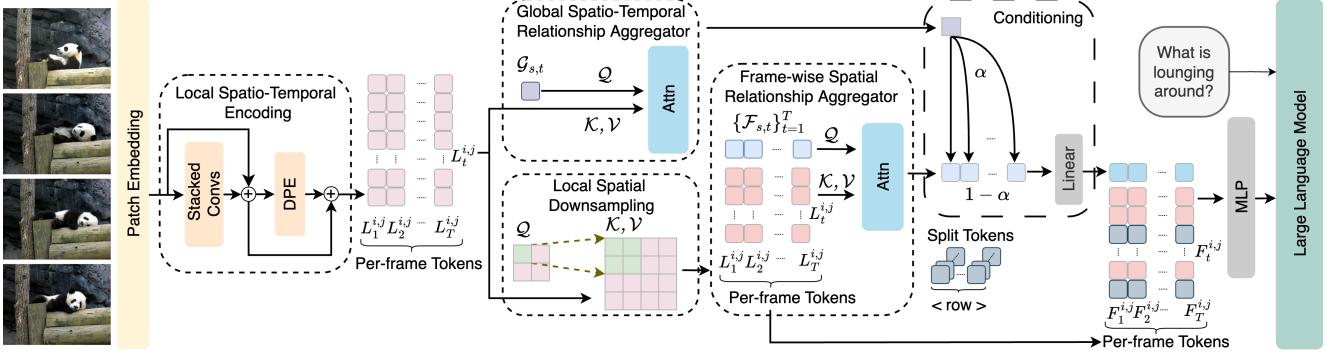
Recent advances in video-language understanding have been primarily driven by extending Large Language Models (LLMs) through various architectural paradigms, as illustrated in Figure 2. Traditional approaches rely heavily on pretrained vision encoders (300M-1.4B parameters) to extract frame-level features, either adapting image encoders for video processing or employing specialized video encoders. These designs typically incorporate additional alignment modules like Q-Former [16] or MLPs to bridge the visual-linguistic gap. While effective, such approaches inherit significant computational overhead from their reliance on heavyweight encoders and complex alignment strategies. In contrast, we propose a radically different design philosophy: an encoder-free architecture that directly processes video inputs through a specialized spatio-temporal alignment block (STAB). This approach

not only eliminates the dependency on pretrained vision encoders but also enables explicit modeling of video dynamics through dedicated spatio-temporal mechanisms. Recent works like Video-ChatGPT [26] and Video-LLaVA [22] have shown strong performance using encoder-based approaches, with some methods further incorporating Q-former modules adapted from BLIP-2 [16] to improve temporal modeling. However, these approaches still fundamentally rely on frame-level features extracted by vision encoders that lack spatio-temporal modeling capabilities.

The aforementioned observations lead us to develop an encoder-free architecture that directly processes video inputs through a specialized spatio-temporal alignment mechanism. In the following sections, we first present our Spatio-Temporal Alignment Block (STAB) (Section 3.1), followed by our training procedure (Section 3.2). Our approach achieves competitive performance with significantly reduced parameters.

#### 3.1. Spatio-Temporal Alignment Block (STAB)

Our Spatio-Temporal Alignment Block (STAB) is illustrated in Figure 3. Since we do not have a pre-trained video encoder where the features need to be aligned with the embedding of the LLM, STAB aligns the video directly with the LLM. A core aspect is the modeling and handling of spatial and temporal information in a video. First, the Local Spatio-Temporal Encoding (LSTE) encodes local information within a small spatio-temporal window. We then separate global context at the video level, which is captured by the Global Spatio-Temporal Relationship Aggregator (GSTRA), and more fine-grained context at the frame level, which is captured by the Frame-wise Spatial Relationship Aggregator (FSRA). While GSTRA models spatio-temporal relationships across the entire video, FSRA mod-



**Figure 3. Detailed architecture of our Spatio-Temporal Alignment Block (STAB):** The input video is first converted into patches. The Local Spatio-Temporal Encoding (LSTE) uses 3D convolutions to model spatio-temporal relations and adds a 3D convolution dynamic position encoding (DPE) to encode position with respect to the local spatio-temporal window. As a result, we obtain per-frame tokens with positional encoding. The tokens are then processed in two ways. While the Global Spatio-Temporal Relationship Aggregator (GSTRA) at the top captures video-level context, the Frame-wise Spatial Relationship Aggregator (FSRA) at the bottom captures spatial context within each frame. To reduce the cost, we perform a Local Spatial Downsampling (LSD) to reduce the spatial dimension for each token. The video-level context tokens and the frame-wise spatial tokens are then linearly combined through learnable weighted fusion ( $\alpha$ ), producing a frame-specific context token. These context tokens are then prepended to their respective frame’s flattened spatial tokens, with  $\langle \text{row} \rangle$  split tokens inserted to demarcate row boundaries in the spatial layout. This combination of global context and preserved spatial structure enables effective video understanding while maintaining computational efficiency.

els spatial relationships across each frame. In our ablation study, we demonstrate that both contain complementary information and their combination improves the results for open-ended video question answering, which includes questions related to the overall video content as well as questions that require a deeper understanding of the temporal information in the video. The corresponding tokens are then fused and aligned with the LLM embedding. We will now discuss each step in detail.

**Patch Embedding:** We first divide each frame of a video  $\mathbf{V} \in \mathbb{R}^{T \times H \times W \times 3}$  with  $T$  frames and spatial resolution  $H \times W$  into non-overlapping patches:

$$\mathbf{P} = f_{\text{patch}}(\mathbf{V}) \in \mathbb{R}^{T \times h \times w \times d} \quad (1)$$

where  $h = H/p$ ,  $w = W/p$  for patch size  $p$ , and  $d$  is the embedding dimension.

**Local Spatio-Temporal Encoding (LSTE):** Videos exhibit rich temporal dynamics that evolve through local spatial neighborhoods. To capture these patterns efficiently, we design a local spatio-temporal aggregation mechanism that processes information within small spatio-temporal windows, allowing each token to aggregate context from its immediate spatial and temporal neighborhood while maintaining computational efficiency:

$$\mathbf{L}'_{st} = \text{Conv3D}_3(\text{Conv3D}_2(\text{Conv3D}_1(\mathbf{P}))) + \mathbf{P} \quad (2)$$

$$\mathbf{L}_{st} = \mathbf{L}'_{st} + \text{DPE}(\mathbf{L}'_{st}) \quad (3)$$

where  $\text{Conv3D}_1$  and  $\text{Conv3D}_3$  uses kernel size  $(1 \times 1 \times 1)$  for channel reduction and restoration,  $\text{Conv3D}_2$  uses kernel size  $(3 \times 1 \times 1)$  for temporal context, and DPE is a

dynamic positional encoder implemented [17] as depthwise 3D convolution that uses kernel size  $(3 \times 3 \times 3)$  for joint spatio-temporal context.

**Local Spatial Downsampling (LSD):** While maintaining fine-grained temporal resolution is crucial for video understanding, the spatial resolution can be efficiently compressed without significant loss of information. We achieve this through a window-based spatial aggregation mechanism that both enriches local spatial context and reduces computational overhead through adaptive downsampling. First, we define the scaled dot-product attention with learnable projection matrices for query, key, and value inputs:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax} \left( \frac{P_q(\mathbf{Q}) P_k(\mathbf{K})^T}{\sqrt{d}} \right) P_v(\mathbf{V}) \quad (4)$$

where  $P_q$ ,  $P_k$ , and  $P_v$  are learnable projection functions for query, key, and value transformations respectively. Then, for each window at position  $(i, j)$  in frame  $t$ :

$$\mathbf{W}_{i,j} \in \mathbb{R}^{1 \times d} \text{ (learnable query)} \quad (5)$$

$$\mathbf{K}_{i,j,t} = \mathbf{L}_{st,t,i:i+2,j:j+2} \in \mathbb{R}^{4 \times d} \text{ (2x2 window)} \quad (6)$$

$$\mathbf{L}_{st,i,j,t}^d = \text{Attn}(\mathbf{W}_{i,j}, \mathbf{K}_{i,j,t}, \mathbf{K}_{i,j,t}) \quad (7)$$

yielding the downsampled representation  $\mathbf{L}_{st}^d \in \mathbb{R}^{T \times h' \times w' \times d}$  where  $h' = h/2$  and  $w' = w/2$ .

**Frame-wise Spatial Relationship Aggregator (FSRA):** Different frames in a video can vary significantly in their spatial content and importance. To capture this frame-specific information effectively, we design a global spatial aggregator that summarizes each frame’s content into a

compact representation while preserving its unique spatial characteristics:

$$\mathbf{W}_t \in \mathbb{R}^{1 \times d} \text{ (frame-specific query)} \quad (8)$$

$$\mathbf{K}_t = \text{Flatten}(\mathbf{L}_{st,t}^d) \in \mathbb{R}^{(h'w') \times d} \text{ (frame tokens)} \quad (9)$$

$$\mathbf{F}_{s,t} = \text{Attn}(\mathbf{W}_t, \mathbf{K}_t, \mathbf{K}_t) \quad (10)$$

producing frame summaries  $\mathbf{F}_s = \{\mathbf{F}_{s,t}\}_{t=1}^T \in \mathbb{R}^{T \times d}$ .

**Global Spatio-Temporal Relationship Aggregator (GSTRA):** Understanding a video often requires capturing both frame-level details and video-wide context. We achieve this through a two-step process: first aggregating global video context, then using this to condition frame-level representations, allowing each frame to maintain its own characteristics while being informed by the broader video context [19, 38]:

$$\mathbf{W}_{st} \in \mathbb{R}^{1 \times d} \text{ (global spatio-temporal query)} \quad (11)$$

$$\mathbf{K}_{st} = \text{Flatten}(\mathbf{L}_{st}) \in \mathbb{R}^{(Th'w') \times d} \text{ (all tokens)} \quad (12)$$

$$\mathbf{G}_{st} = \text{Attn}(\mathbf{W}_{st}, \mathbf{K}_{st}, \mathbf{K}_{st}) \in \mathbb{R}^{1 \times d} \quad (13)$$

Then, we condition each frame summary with the global context:

$$\mathbf{F}_{r,t} = f_{\text{proj}}(\alpha \mathbf{F}_{s,t} + (1 - \alpha) \mathbf{G}_{st}) \quad (14)$$

where  $\alpha \in \mathbb{R}^d$  is learned and  $f_{\text{proj}}$  is a shared linear projection, yielding the final aggregated representation  $\mathbf{F}_r \in \mathbb{R}^{T \times d}$ .

**Final Token Sequence Construction:** To maintain the structural information of the video while providing both local and global context to the language model, we carefully construct the final token sequence. First, we organize the downsampled tokens  $\mathbf{L}_{st}^d$  into frame-wise sequences with row delimiter tokens:

$$\mathbf{F}_{\text{tokens}} = \{\text{Add-Row-Split}(\mathbf{L}_{st,t}^d)\}_{t=1}^T. \quad (15)$$

Here, we insert row-split tokens  $\langle \text{row} \rangle$  after each row of spatial tokens in a frame, helping maintain 2D spatial structure. Then, we combine these with the conditioned frame representations:

$$\mathbf{F} = \{[\mathbf{F}_{r,t}; \mathbf{F}_{\text{tokens},t}]\}_{t=1}^T \quad (16)$$

placing each frame’s conditioned representation  $\mathbf{F}_{r,t}$  at the start of its corresponding token sequence. The final visual tokens are obtained by flattening  $\mathbf{F}$  and projecting through an MLP to the language model’s embedding space:

$$\mathbf{V}_{\text{tokens}} = \text{MLP}(\text{Flatten}(\mathbf{F})) \quad (17)$$

while maintaining spatial structure through the split tokens.

### 3.2. Training Procedure

Our training strategy follows three carefully designed stages to ensure stable optimization and effective knowledge transfer. First, we define two core losses used throughout training. For visual alignment, we use a distillation loss that aligns our predicted visual tokens with a teacher model. Let  $\mathbf{O} = (o_1, \dots, o_N) \in \mathbb{R}^{N \times d}$  be the output tokens from a video-language model, where  $N$  is the sequence length and  $d$  is the embedding dimension. We can partition these tokens into visual and non-visual tokens:

$$\mathbf{O} = [\mathbf{V}_{\text{pred}}; \mathbf{O}_{\text{other}}] \quad (18)$$

where  $\mathbf{V}_{\text{pred}} \in \mathbb{R}^{M \times d}$  represents the  $M$  predicted visual tokens and  $\mathbf{O}_{\text{other}}$  contains the remaining textual and special tokens. Given an input video, let  $\mathbf{V}_{\text{teacher}} \in \mathbb{R}^{M \times d}$  be the visual tokens extracted from a pre-trained teacher model. The individual predicted tokens and teacher tokens are denoted as  $\mathbf{v}_{\text{pred},i}$  and  $\mathbf{v}_{\text{teacher},i}$ , respectively. The distillation loss is then defined as:

$$\mathcal{L}_{\text{distill}} = -\frac{1}{M} \sum_{i=1}^M \frac{\mathbf{v}_{\text{pred},i}^\top \mathbf{v}_{\text{teacher},i}}{\|\mathbf{v}_{\text{pred},i}\| \|\mathbf{v}_{\text{teacher},i}\|}. \quad (19)$$

We use LanguageBind [45] as our teacher model for extracting  $\mathbf{V}_{\text{teacher}}$ . For text generation, we use a standard cross-entropy loss over the textual tokens in  $\mathbf{O}_{\text{other}}$ :

$$\mathcal{L}_{\text{text}} = -\sum_{t \in \mathcal{T}} \log p(o_t | o_{<t}, \mathbf{O}) \quad (20)$$

where  $\mathcal{T}$  denotes the indices of textual tokens in  $\mathbf{O}_{\text{other}}$ , and  $p(o_t | o_{<t}, \mathbf{O})$  is the model’s prediction probability for token  $o_t$  given previous tokens.

**Initial Alignment Stage:** The primary goal of this stage is to establish a foundational understanding of video content through our spatio-temporal alignment mechanism. Using 351k video-text pairs from WebVid [4] (half of the full dataset), we train only the STAB components while keeping the LLM frozen. This stage is crucial for preventing training collapse and ensuring stable convergence, as it allows the alignment components to develop basic video understanding capabilities without interfering with the LLM’s pre-trained knowledge.

**Visual-Language Integration Stage:** Using the same WebVid dataset, we fine-tune the entire model end-to-end to develop joint visual-language understanding. This stage allows the LLM to adapt its language understanding capabilities to video-specific contexts while maintaining its general language abilities.

**Instruction Tuning Stage:** The final stage focuses on developing instruction-following capabilities while maintaining strong video understanding. Using the Video-ChatGPT [26] instruction dataset (100k samples), we fine-tune the model with emphasis on generating appropriate responses to video-based instructions and questions.

Model	Vision Size	Modality	Pretrain Data	Finetune Data	MSVD-QA		MSRVTT-QA		TGIF-QA*		Activity Net-QA		
					Accuracy	Score	Accuracy	Score	Accuracy	Score	Accuracy	Score	
<i>Encoder-based Vision-Language Models</i>													
Different Datasets	LLaMA Adapter [43]	404.3M	I	567K	52K	54.9	3.1	43.8	2.7	-	-	34.2	2.7
	VideoChat [18]	1.2B	V	25M	18K	56.3	2.8	45.0	2.5	21.3	1.9	26.5	2.2
	Video-LLaMA [42]	1.1B	V	3.1M	164K	51.6	2.5	29.6	1.8	-	-	12.4	1.1
	ChatUniVi [15]	307M	V+I	1.6M	649K	65.0	3.6	54.6	3.1	38.2	3.0	45.8	3.2
	LLaMA-VID [21]	1B	V+I	790K	763K	69.7	3.7	57.7	3.2	-	-	47.4	3.3
	Video-LLaVA [22]	425M	V+I	1.26M	765K	70.7	3.9	59.2	3.5	47.0	3.3	45.3	3.3
	VideoChat2 [20]	496M	V+I	37M	2M	70.0	3.9	54.1	3.3	-	-	49.1	3.3
<i>Encoder-free Vision-Language Models</i>													
Same Dataset	EVE [11]	30M	I	33M	665K	14.3	1.7	-	-	-	-	6.8	1.8
	<i>Encoder-based Vision-Language Models</i>												
Same Dataset	Video-ChatGPT [26]	307M	V	-	100K	64.9	3.3	49.3	2.8	40.7	3.1	35.2	2.8
	Video-LLaVA [22]	425M	V	702K	100K	64.8	-	58.3	-	41.7	-	40.7	-
<i>Encoder-free Vision-Language Models</i>													
Same Dataset	EVE*	30M	V	702K	100K	60.5	3.3	49.7	3.0	39.2	2.9	38.1	3.0
	Video-Panda (ours)	45M	V	702K	100K	64.7	3.8	54.8	3.4	42.9	3.2	40.0	3.3

Table 1. Comparison with other video-language models that use LLMs with 7B parameters on open-ended video question answering. *Vision Size* refers to #parameters of vision encoder and alignment modules. *Modality* indicates whether videos and/or images are used as training data. For *TGIF-QA\**, we re-evaluated the results since the performance depends on the current version of GPT-3.5 which changes over time and highly impacts the evaluation. *EVE\** is our extension of EVE [11] to video data.

## 4. Experiments

### 4.1. Datasets and Implementation Details

**Training Data:** Our training dataset is organized into three distinct stages. The first stage uses a randomly sampled subset of 351K video-text pairs (50%) from Valley [25], which contains 702K video-text pairs sourced from the WebVid [4] dataset. Stage two expands to the full Valley dataset of 702K pairs. The final stage incorporates an additional 100K video-text instruction dataset from Video-ChatGPT [26], focusing on complex visual reasoning tasks. **Open-Ended Video QA Evaluation Data:** We evaluate our approach on four open-ended VideoQA datasets: MSRVTT-QA [40], MSVD-QA [9], TGIF-QA [14], and ActivityNet-QA [41]. MSRVTT-QA and MSVD-QA include five question types (what, who, how, when, and where), while TGIF-QA uses four types (object, number, color, and location). MSRVTT-QA contains 10K video clips and 243K question-answer pairs (158K/12K/73K for training/validation/testing), while MSVD-QA includes 1,970 video clips and 51K question-answer pairs (32K/6K/13K split). TGIF-QA comprises 46K GIFs and 53K question-answer pairs (39K/13K for training/testing). ActivityNet-QA, featuring longer videos averaging three minutes, covers nine question types (motion, spatial, temporal, yes-no, color, object, location, number, and other) with 5,800 videos and 58K question-answer pairs (32K/18K/8K split). We implemented a zero-shot evaluation protocol utilizing GPT-assisted assessment to analyze the model’s performance. The evaluation methodology uses both prediction accuracy and a 5-point scale assessment.

**Fine-Grained Video QA Evaluation Data:** For a more fine-grained evaluation and analysis of our method, we

Model	Correctness	Detail	Context	Temporal	Consistency	AVG	
						Encoder-based Vision-Language Models	Encoder-free Vision-Language Models
<i>Different Datasets</i>							
Different Datasets	VideoChat [18]	2.23	2.50	2.53	1.94	2.24	2.29
	LLaMA Adapter [43]	2.03	2.32	2.30	1.98	2.15	2.16
	Video-LLaMA [42]	1.96	2.18	2.16	1.82	1.79	1.98
	ChatUniVi [15]	2.89	2.91	3.46	2.39	2.81	2.89
	LLaMA-VID [21]	2.96	3.00	3.53	2.46	2.51	2.89
	Video-LLaVA [22]	2.84	2.86	3.44	2.46	2.57	2.81
	VideoChat2 [20]	3.02	2.88	3.51	2.66	2.81	2.98
<i>Same Datasets</i>							
Same Datasets	Video-ChatGPT [26]	2.40	2.52	2.62	1.98	2.37	2.38
	Video-LLaVA* [22]	2.46	2.37	2.89	2.12	2.17	2.40
<i>Encoder-free Vision-Language Models</i>							
Video-Panda (ours)	2.74	2.47	3.01	2.26	2.36	2.57	

Table 2. Comparison of video-language models on fine-grained video question answering metrics (scale 1-5) across correctness, detail, context, temporal reasoning, and consistency. *Video-LLaVA\**: trained with video-only datasets for fair comparison.

adopt the benchmark introduced by Video-ChatGPT [26]. Built upon the ActivityNet-200 [8] dataset, this framework assesses responses across five dimensions: correctness of information, detail orientation, contextual understanding, temporal understanding, and consistency. Each aspect is scored on a scale of 1-5, evaluating factual accuracy, response completeness, context alignment, sequential comprehension, and reliability across similar queries.

**Implementation Details:** Our model is built on Vicuna-7B v1.5 [10] tailored to vision-language applications. The teacher model is LanguageBind [45], which is initialized from ViT-L/14 [12] and pre-trained with contrastive objectives to align multi-modal representations within a unified embedding space. Eight frames are sampled uniformly from each video, and the maximum image/frame width or height is limited to 448 pixels. Input to the teacher model is padded and resized to 224x224. Training samples for the three stages are 351K, 702K, and 100K, respectively. For

Model	#Param.(M)	Inference time (ms)
VideoChatGPT [26]	307	171
Video-LLaVA [22]	425	125
Video-Panda	45	41

Table 3. Comparison of the number of parameters and inference time of the vision part.

each training batch, cross-attentions are performed between query and valid spatial tokens by removing padded tokens for efficient modeling. We keep other hyperparameters the same as previous work [11]. The full hyperparameter settings can be found in the supplementary material.

## 4.2. Quantitative Evaluation

In this subsection, we present a comprehensive evaluation of our model’s performance. We analyze the open-ended and fine-grained video understanding capabilities across multiple benchmark datasets, comparing our encoder-free approach with approaches trained on the same dataset as well as approaches that use more or other data for training. Subsequently, we examine the computational efficiency through parameter count and inference time.

**Open-Ended Video Question Answering:** The results of our evaluation for open-ended video question answering are presented in Table 1. Our proposed Video-Panda model, despite being the only encoder-free video-language model, demonstrates competitive performance with encoder-based approaches when compared to methods that have been trained on the same dataset (bottom). We first compare the results of Video-Panda with Video-ChatGPT [26]. While Video-ChatGPT achieves a slightly higher accuracy on MSVD-QA (64.7% vs 64.9%), Video-Panda achieves a higher score of 3.8. For the other datasets MSRVTT-QA, TGIF-QA, and Activity Net-QA, Video-Panda outperforms Video-ChatGPT for all metrics. In comparison to Video-LLaVA [22], which has even more parameters than Video-ChatGPT, Video-Panda achieves competitive results and even outperforms it on TGIF-QA. Since EVE [11] performs poorly on the video benchmarks, we also extended EVE to the video domain (see supplementary material). Extending EVE to the video domain, however, performs substantially worse than Video-Panda. This shows the importance of the proposed video-specific alignment module. For completeness, we also compare to methods that have been pre-trained and fine-tuned on other datasets although the results are not directly comparable. Nevertheless, Video-Panda also outperforms VideoChat [18] and Video-LLaMA [42], which are trained only on video as well.

**Fine-Grained Video Question Answering:** The results for fine-grained video question answering are presented in Table 2. Our proposed Video-Panda model, despite being an encoder-free approach with only 45M parameters as shown

Model	MSVD-QA	Activity Net-QA
<i>Spatial</i>		
w/o <row>	63.2/3.7	39.5/3.3
w/o FSRA	63.4/3.7	39.2/3.3
w/o LSD (avg pool)	58.0/3.6	38.1/3.2
<i>Temporal</i>		
w/o LSTE	63.6/3.7	39.4/3.3
w/o GSTRA	63.0/3.7	38.2/3.2
w/o GSTRA & LSTE	62.2/3.7	38.1/3.2
Video-Panda	<b>64.7/3.8</b>	<b>40.0/3.3</b>

Table 4. Ablation study on the impact of removing different spatial and temporal modules used in our design.

Distillation Loss	MSVD-QA	Activity Net-QA
w/o Distillation	63.1/3.7	39.8/3.3
Mean Squared Error	63.5/3.7	38.2/3.2
Negative Cosine Similarity	<b>64.7/3.8</b>	<b>40.0/3.3</b>

Table 5. Ablation study on the impact of not using distillation loss or a different (MSE) loss.

in Table 1, demonstrates strong performance when compared to methods trained on the same dataset. Video-Panda outperforms Video-LLaVA in all aspects. When compared to Video-ChatGPT, Video-Panda shows substantial improvements in correctness (2.74 vs 2.40), context (3.01 vs 2.62), and temporal understanding (2.26 vs 1.98), although it achieves slightly lower scores in detail and consistency. It is worth noting that Video-ChatGPT processes 100 frames per video while Video-Panda uses only 8 frames, making our model’s performance particularly impressive. We also compare to approaches that have been pre-trained and fine-tuned on different datasets.

**Parameters and Inference Time:** As shown in Table 3, our Video-Panda model requires much fewer parameters (45M) for its visual component compared to VideoChatGPT (307M) and Video-LLaVA (425M). This also translates directly to better inference speed, which is very important for practical applications. Video-Panda processes videos in 41ms, approximately 4× faster than VideoChatGPT (171ms) and 3× faster than Video-LLaVA (125ms). These results demonstrate that our encoder-free approach not only maintains competitive performance but also offers significant computational advantages.

## 5. Analysis and Ablations

To evaluate our design choices and their impact on model performance, we conduct ablation studies regarding architectural components, distillation loss, and qualitative impact

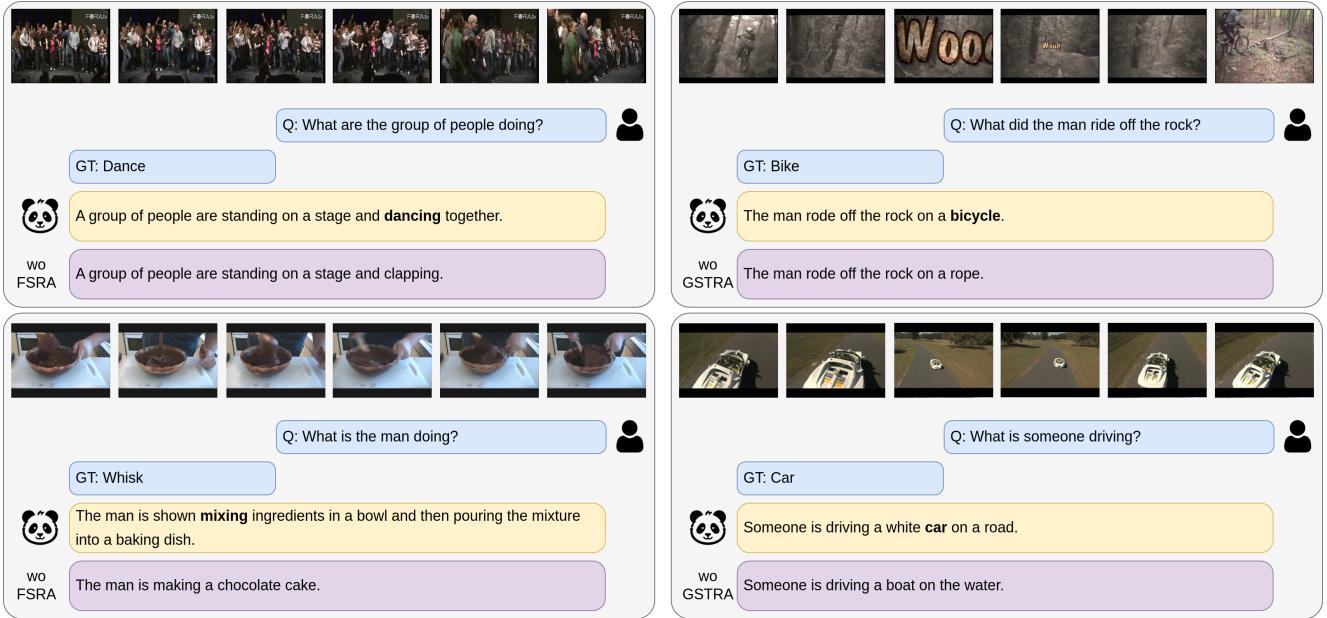


Figure 4. Qualitative examples showing the impact of removing Frame-wise Spatial Relationship Aggregator (FSRA) and Global Spatio-Temporal Relationship Aggregator (GSTRA).

of the spatial and temporal aggregators. Additional ablations are presented in the supplementary materials.

**Component Analysis:** We first analyze the contribution of each architectural component in [Table 4](#). The removal of row tokens ( $\langle \text{row} \rangle$ ) leads to a noticeable drop in performance, particularly on MSVD-QA (1.5% decrease), highlighting the importance of maintaining spatial structure information. Similarly, removing the Frame-wise Spatial Relationship Aggregator (FSRA) results in a decrease of 1.3%. Replacing Local Spatial DownSampling (LSD) by simply average pooling decreases the performance on MSVD-QA by 6.7%. The Local Spatio-Temporal Encoding (LSTE) and Global Spatio-Temporal Relationship Aggregator (GSTRA) modules show complementary benefits, with their combined removal resulting in a 2.5% drop on MSVD-QA, indicating the effectiveness of our spatio-temporal modeling approach.

**Distillation Loss:** The choice of the distillation loss function shows some impact on the model performance, as shown in [Table 5](#). Our negative cosine similarity loss outperforms MSE by 1.2% on MSVD-QA and 1.8% on ActivityNet-QA. Removing distillation entirely still maintains relatively strong performance, particularly on ActivityNet-QA. This suggests that while our distillation approach provides meaningful benefits, the architectural design itself captures significant video understanding capabilities. The superior performance of cosine similarity over MSE indicates the importance of direction-based alignment rather than exact magnitude matching in feature space.

**Qualitative Ablations:** The qualitative examples in [Figure 4](#) demonstrate the critical role of our FSRA and GSTRA components in accurate visual understanding. Without FSRA, the model struggles to properly interpret more granular actions, as evidenced by misinterpreting a coordinated dance performance as mere “clapping” and failing to recognize the “mixing” action in favor of the end goal of “making a chocolate cake”. When removing GSTRA, the model tends to focus on single frames instead of considering the context of the entire video, which can be misleading. For instance, the construction at the end of the video is misinterpreted as “rope” and the “car” on a “road” is misinterpreted as a “boat” on the “water” at the beginning. This shows that the global video context from GSTRA as well as the frame-wise representation from FSRA are needed.

## 6. Conclusion

This work introduces an efficient encoder-free approach to video-language understanding through our Spatio-Temporal Alignment Block (STAB). We reduce visual processing parameters by  $6.5\times$  or greater while maintaining competitive performance or outperforming comparable approaches in open-ended video question answering, obtaining 64.7% accuracy on MSVD-QA and 54.8% on MSRVTT-QA, as well as fine-grained video understanding. Additionally, STAB processes videos  $3\text{-}4\times$  faster than encoder-based methods. Our method shows promise for making video-language models more efficient, potentially benefiting longer videos and other tasks.

## References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1, 2
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NIPS*, 2022. 11
- [3] Anthropic. Anthropic: Introducing the next generation of claude. <https://www.anthropic.com/news/claude-3-family>, 2024. 2
- [4] Max Bain, Arsha Nagrani, Gülcin Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 5, 6, 11
- [5] Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Soman, and Sağnak Taşlılar. Introducing our multimodal models. <https://www.adept.ai/blog/fuyu-8b>, 2023. 2
- [6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 2
- [7] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023. 2
- [8] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 6, 12
- [9] David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 2, 6
- [10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhang-hao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yong-hao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%\* chatgpt quality. <https://lmsys.org/blog/2023-03-30-vicuna>, 2023. 2, 6
- [11] Haiwen Diao, Yufeng Cui, Xiaotong Li, Yueze Wang, Huchuan Lu, and Xinlong Wang. Unveiling encoder-free vision-language models. In *NeurIPS*, 2024. 2, 6, 7, 12
- [12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 6
- [13] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023. 2
- [14] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 6
- [15] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *CVPR*, 2024. 1, 3, 6
- [16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 2, 3
- [17] Kunchang Li, Yali Wang, Gao Peng, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatial-temporal representation learning. In *ICLR*, 2022. 4
- [18] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhui Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 1, 2, 6, 7
- [19] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Limin Wang, and Yu Qiao. Uniformerv2: Unlocking the potential of image vits for video understanding. In *ICCV*, 2023. 5
- [20] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mybench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024. 2, 6
- [21] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *ECCV*, 2025. 2, 6
- [22] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning unified visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023. 1, 3, 6, 7
- [23] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 11
- [24] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 2
- [25] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Minghui Qiu, Pengcheng Lu, Tao Wang, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*, 2023. 6, 11
- [26] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *ACL*, 2024. 1, 2, 3, 5, 6, 7, 12
- [27] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Videogpt+: Integrating image and video encoders for enhanced video understanding. *arXiv preprint arxiv: 2406.09418*, 2024. 1

- [28] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024. 2
- [29] OpenAI. Openai: Introducing chatgpt. <https://openai.com/index/chatgpt>, 2022. 2
- [30] OpenAI. Gpt-4v(ision) system card. *OpenAI*, 2023. 2
- [31] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022. 2
- [32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 12
- [33] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca), 2023. 2
- [34] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 2
- [35] Hugo Touvron, Thibaut Lavrille, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1, 2
- [36] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwala Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2
- [37] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *ECCV*, 2024. 1
- [38] Syed Talal Wasim, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Vita-clip: Video and text adaptive clip via multimodal prompting. In *CVPR*, 2023. 5
- [39] xAI. xai: Grok-1.5 vision preview. <https://x.ai/blog/grok-1.5v>, 2024. 2
- [40] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 6
- [41] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueteng Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, 2019. 6
- [42] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 1, 3, 6, 7
- [43] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. In *ICLR*, 2024. 1, 6
- [44] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023. 1, 2
- [45] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiaxi Cui, HongFa Wang, Yatian Pang, Wenhai Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*, 2023. 5, 6
- [46] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*, 2024. 2

## Appendix

This section contains supplemental material, offering further results and analysis to complement the main paper. We provide additional details on the following topics:

- Detailed Hyperparameters ([Appendix A](#))
- Additional Ablations ([Appendix B](#))
- Additional Dataset Details ([Appendix C](#))
- Additional Qualitative Ablations ([Appendix D](#))
- EVE Baseline for Videos ([Appendix E](#))
- Broader Impact ([Appendix F](#))

### A. Detailed Hyperparameters

In [Table 6](#) we provide comprehensive hyperparameter configurations for Video-Panda’s three-stage training process.

Hyperparameter	Stage-1	Stage-2	Stage-3
Batch Size	2048	2048	1024
Learning Rate (lr)	4e-4	4e-5	2e-5
LR Schedule		cosine decay	
LR Warmup Ratio	0.03	0.01	0.01
Weight Decay	0	0	0
Epoch	1	1	1
Optimizer	AdamW	AdamW	AdamW
DeepSpeed Stage	2	2	2
LLM	Frozen	Trainable	Trainable
STAB	Trainable	Trainable	Trainable

Table 6. Hyperparameter Settings

### B. Additional Ablations

**Training Data for Initial Alignment:** [Table 7](#) shows the impact of data scale during initial alignment. Using the full dataset (702K samples) in Stage 1 yields marginally lower performance compared to using half (351K samples). This suggests our staged training approach benefits from gradual complexity scaling, allowing the model to establish robust representations before incorporating the complete dataset in later stages.

**Downsampling Position:** Regarding temporal layer placement in [Table 8](#), we find that applying LSD after LSTE improves performance on all datasets except of ActivityNet-QA. For consistency across our experiments, we maintain LSD placement after LSTE.

**Downsampling Strategy:** As shown in [Table 9](#), our LSD method outperforms alternative approaches. The Perceiver Resampler (PR) performs notably poorly (21.3% lower on MSVD-QA), likely due to excessive information compression. While average pooling performs better, it still underperforms LSD by 6.7%, demonstrating the superiority of our learnable downsampling approach.

#Samples for Initial Alignment	MSVD-QA	Activity Net-QA
702K Video-Text Pairs (full)	63.7/3.8	39.7/3.3
351K Video-Text Pairs (half)	<b>64.7/3.8</b>	<b>40.0/3.3</b>

Table 7. Ablation study on amount of data for the first training stage.

Model	MSVD-QA	MSRVTT-QA	TGIF-QA	Activity Net-QA
Before LSTE	64.2/3.8	54.6/3.4	42.7/3.2	<b>42.3/3.3</b>
After LSTE (Ours)	<b>64.7/3.8</b>	<b>54.8/3.4</b>	<b>42.9/3.2</b>	40.0/3.3

Table 8. Ablation study on downsampling positions of LSD.

Model	MSVD-QA	Activity Net-QA
w/o LSD (half-resolution)	48.2/3.3	38.5/3.2
w/o LSD (avg pool)	58.0/3.6	38.1/3.2
w/o LSD (PR)	43.4/3.2	27.8/2.9
Video-Panda (LSD)	<b>64.7/3.8</b>	<b>40.0/3.3</b>

Table 9. Ablation study on downsampling methods. PR stands for Perceiver Resampler [2].

Model	MSVD-QA	Activity Net-QA
CLIP	60.3/3.5	38.6/3.2
InternVideo	62.5/3.6	39.6/3.2
DINOv2	61.7/3.5	38.1/3.2
LanguageBind (Video-Panda)	<b>64.7/3.8</b>	<b>40.0/3.3</b>

Table 10. Ablation study different teacher encoders.

**Different Teachers:** As shown in [Table 10](#), LanguageBind consistently outperforms other teacher encoders across both datasets. While InternVideo achieves the second-best performance, it still falls short by 2.2% on MSVD-QA and 0.4% on Activity Net-QA. CLIP and DINOv2 show comparable performance to each other but lag behind LanguageBind by 3-4%, demonstrating the effectiveness of our chosen teacher encoder.

### C. Additional Dataset Details

**Pre-training Dataset:** The Valley-Pretrain-702K dataset is a large-scale pre-training dataset designed for video-language understanding tasks. It comprises 702K video-text pairs from the WebVid dataset [4], filtered by [25] using methods established by LLaVA [23] to optimize the balance between conceptual diversity and training efficiency. The dataset is structured as single-round dialogues, where each video is paired with questions about its content and corresponding caption-based answers.

**Fine-tuning Dataset:** The Video-ChatGPT-100K dataset was developed for fine-tuning video-language models, comprising 100K video instruction samples collected

by [26]. The dataset combines human expertise with semi-automated methods to balance quality and scalability. Expert annotators provide detailed, context-rich descriptions that enhance the model’s comprehension of complex video content. A semi-automatic framework leverages state-of-the-art vision-language models to generate large-scale annotations efficiently, ensuring substantial data volume while maintaining rigorous quality standards.

**Fine-Grained Video QA Evaluation Dataset:** We evaluate fine-grained video question answering using the Video-based Text Generation Performance Benchmarking methodology developed by Video-ChatGPT [26]. This benchmark provides a comprehensive evaluation framework for assessing text generation in video-based conversational models. Using the ActivityNet-200 dataset [8], which contains videos with descriptive captions and human-annotated question-answer pairs, the framework implements a systematic evaluation approach. The methodology utilizes GPT-3.5 to evaluate models across multiple dimensions on a scale of 1 to 5. The assessment criteria include:

- (i) *Correctness of Information*: Evaluates the accuracy of generated text and its alignment with video content.
- (ii) *Detail Orientation*: Assesses response comprehensiveness, examining both coverage of major points and specificity of details.
- (iii) *Contextual Understanding*: Measures the model’s ability to interpret and respond within the video’s broader context.
- (iv) *Temporal Understanding*: Evaluates the model’s capacity to track and articulate the chronological sequence of events.
- (v) *Consistency*: Assesses the model’s ability to maintain coherent responses across different questions and video segments.

## D. Additional Qualitative Ablations

We present additional qualitative examples of our ablation studies in [Figure 5](#), demonstrating Video-Panda’s effectiveness across various video understanding tasks. When using the complete training dataset in Stage 1 (left-top example), the model exhibits overfitting tendencies due to data imbalance, as evidenced by the example showing dog interactions—likely influenced by the disparity between dog (7,807) and cat (5,050) instances in the Valley dataset. The right-top example reveals that placing the LSD module before LSTE impairs cliff recognition due to early token downsampling and information loss, supporting our decision to maintain LSD placement after LSTE. Models using alternative approaches (average pooling, half resolution, or perceiver resampler) struggle with content recognition (e.g., cucumber, cat, pandas) compared to our learnable downsampling approach. Additionally, models using image-

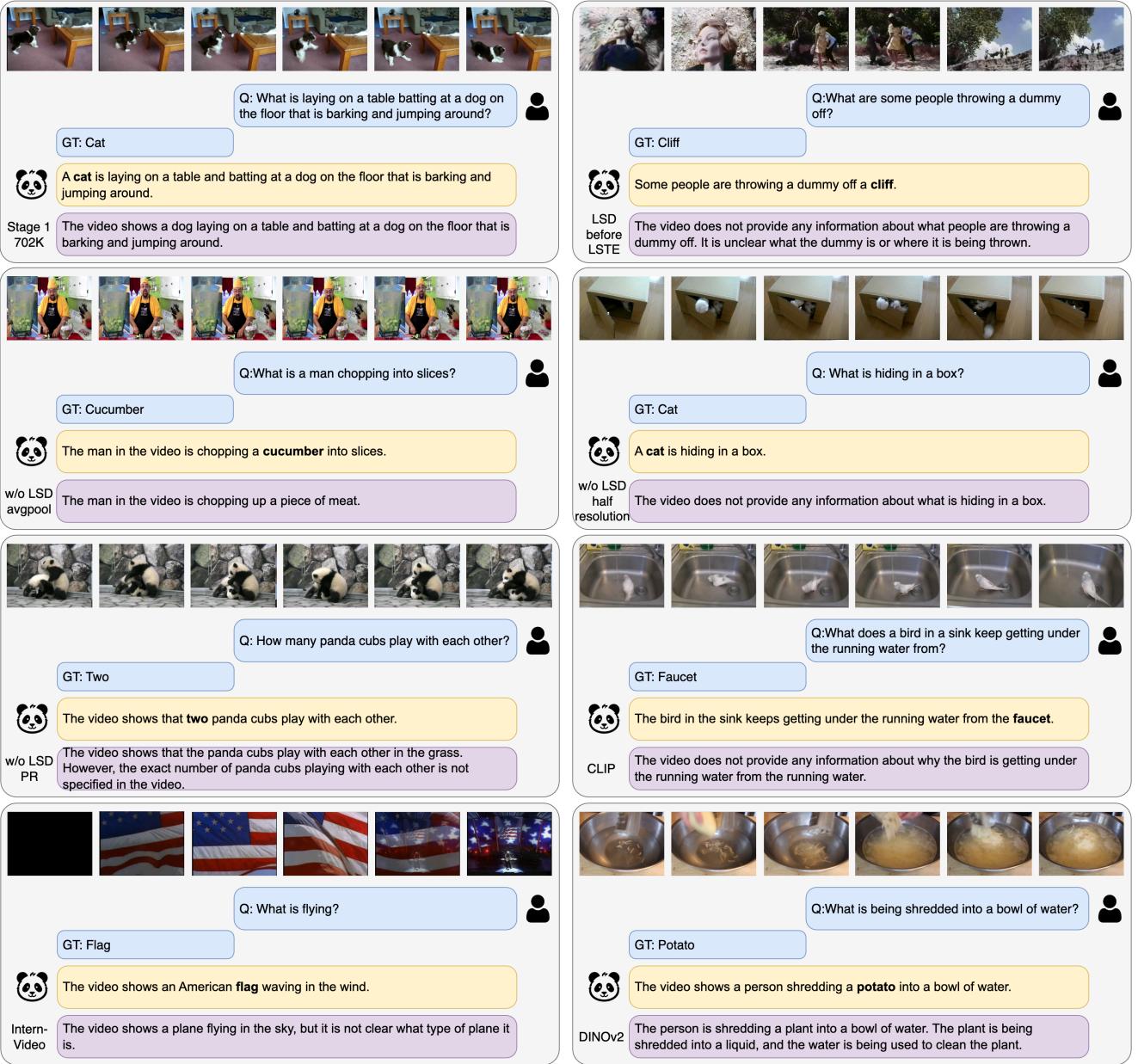
based teachers (CLIP and DINOv2) tend to make frame-specific predictions rather than considering global context, as demonstrated by their failure to recognize shredded potatoes across multiple frames.

## E. EVE Baseline for Videos

As the original EVE model [11] was designed for image processing, we conducted a fair comparison by re-training it (denoted as *EVE\** in [Table 1](#)) using identical video data (Valley-Pretrain-702K and Video-ChatGPT-100K datasets). For processing videos through *EVE\**, each frame was treated independently as a separate image, with CLIP-ViT-L/14 [32] serving as the teacher model for distillation. While this approach enables frame-level analysis, it neglects temporal relationships. In our implementation, we employ Learnable Selective Downsampling (LSD) to process video frames efficiently, reducing each frame to a consistent token count while preserving essential information. The resulting tokens are flattened into a single sequence, with special split tokens inserted between frame representations to maintain frame boundaries and enable temporal relationship learning.

## F. Broader Impact

We introduce Video-Panda, an encoder-free Video Language Model for video understanding. Our model addresses key ethical and practical challenges in large-scale AI deployment. While many VLMs raise concerns about data bias, privacy, and computational costs, Video-Panda mitigates these issues through two key design choices: training exclusively on publicly available datasets and eliminating the need for a pretrained encoder. This approach not only reduces ethical concerns but also significantly lowers computational requirements and deployment costs, making the model more accessible and environmentally sustainable.



**Figure 5. Comparative analysis of Video-Panda qualitative ablation studies:** The figure presents eight video examples with ground truth (GT) annotations and model predictions under different training configurations. The top row demonstrates: the effect of 702K training samples in stage 1, and the impact of performing Local Spatial Downsampling (LSD) before Local Spatial-Temporal Encoding (LSTE). The second row shows results from removing LSD while using: average pooling, half-resolution and perceiver resampler (third row left). The third row right and bottom row illustrate the effects of different teacher models: using Intern-Video and CLIP, and utilizing DINOv2 for knowledge distillation. Each example includes the original model prediction (yellow) and an ablated version (purple), highlighting how architectural and training choices affect Video-Panda’s ability to interpret dynamic visual scenes and answer questions accurately.