Unifying Specialized Visual Encoders for Video Language Models

Jihoon Chung [*1] **Tyler Zhu** [*1] **Max Gonzalez Saez-Diez** [1]
Juan Carlos Niebles [2] **Honglu Zhou** [2] **Olga Russakovsky** [1]

Abstract

Recent advances in vision backbones have yielded powerful and diverse visual and video encoders. Yet, current Video Large Language Models encode visual inputs using an encoder from a single backbone family, limiting the amount and type of visual information they can process. We propose MERV, a Multi-Encoder Video Representation, which utilizes multiple encoders for a comprehensive video representation. To optimize heterogeneous features from a broad spectrum of encoders and ensure efficient and coherent feature integration, MERV first aligns encoder features spatio-temporally, then projects them into a unified structure, and finally fuses them through cross-attention. Under fair comparison, MERV achieves up to 4.62% higher accuracy than its base model, while introducing minimal extra parameters and training faster than equivalent single-encoder methods after parallelizing visual processing. Qualitative analysis shows MERV successfully captures and integrates domain knowledge from each encoder, opening new possibilities for scaling enhanced video understanding.

1. Introduction

Inspired by the sophisticated reasoning abilities of recent Large Language Models (LLMs) (Chiang et al., 2023; Chowdhery et al., 2023; OpenAI, 2023), researchers have focused on using them in many other domains to great success. The video counterparts, known as Video Large Language Models (VideoLLMs) (Bain et al., 2021; Li et al., 2023c; Lin et al., 2024; Luo et al., 2023; Maaz et al., 2024; Yu et al., 2024b), connect pre-trained vision encoders to LLMs by training a modality bridge from the vision space to the language space, allowing for reasoning to happen in the highly expressive language domain.

[*]Equal contribution [1]Department of Computer Science, Princeton University, Princeton, NJ, United States [2]Salesforce Research, Palo Alto, CA, United States. Correspondence to: Jihoon Chung, Tyler Zhu <{jc5933,tylerzhu}@princeton.edu>.

Proceedings of the 42nd International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

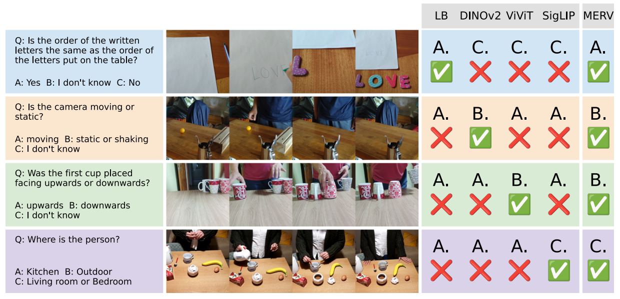

LB
DINOv2
ViViT
SigLIP
MERV
Q: Is the order of the written letters the same as the order of the letters put on the table?
A: Yes B: I don't know C: No
A. ✓ C. ✗ C. ✗ C. ✗ A. ✓
Q: Is the camera moving or static?
A: moving B: static or shaking C: I don't know
A. ✗ B. ✓ A. ✗ A. ✗ B. ✓
Q: Was the first cup placed facing upwards or downwards?
A: upwards B: downwards C: I don't know
A. ✗ A. ✗ B. ✓ A. ✗ B. ✓
Q: Where is the person?
A: Kitchen B: Outdoor C: Living room or Bedroom
A. ✗ A. ✗ A. ✗ C. ✓ C. ✓

Figure 1. Examples where a single encoder model is the only model to correctly answer Perception Test questions (Pătrăucean et al., 2023), while MERV can correctly answer all types.

Most multimodal LLMs, such as LLaVA (Liu et al., 2023) for images and Video-LLaVA (Lin et al., 2024) for videos, opt for contrastively pre-trained encoders like CLIP (Radford et al., 2021) and LanguageBind (Zhu et al., 2024a). Their vision-language pre-training naturally lends itself as a bridge between the vision input and the LLM, circumventing the need to train heavy vision-language alignment modules like a QFormer (Li et al., 2022). These encoders are almost always pre-trained separately and vary in architecture, training data, and optimization strategy. Consequently, the features extracted by these encoders exhibit unique characteristics, each with inherent strengths and limitations. Contrastive encoders like CLIP may be better suited with their multimodal semantic alignment, but are inferior to models such as DINOv2 (Oquab et al., 2023) at fine-grained object level understanding. They also fail to take advantage of models trained specifically on videos, such as ViViT (Arnab et al., 2021). Despite this clear tension between vision backbones, previous research in VideoLLMs has relied on *only* one vision encoder for visual processing as one was thought to be sufficient for visual understanding, and already difficult enough to achieve vision-language alignment with. Any more encoders was unnecessary and not an effective tradeoff of runtime for compute.

In this paper, we argue that this choice to not use multiple encoders in existing VideoLLMs unnecessarily restricts their capabilities. For example, in Figure 1 we can see

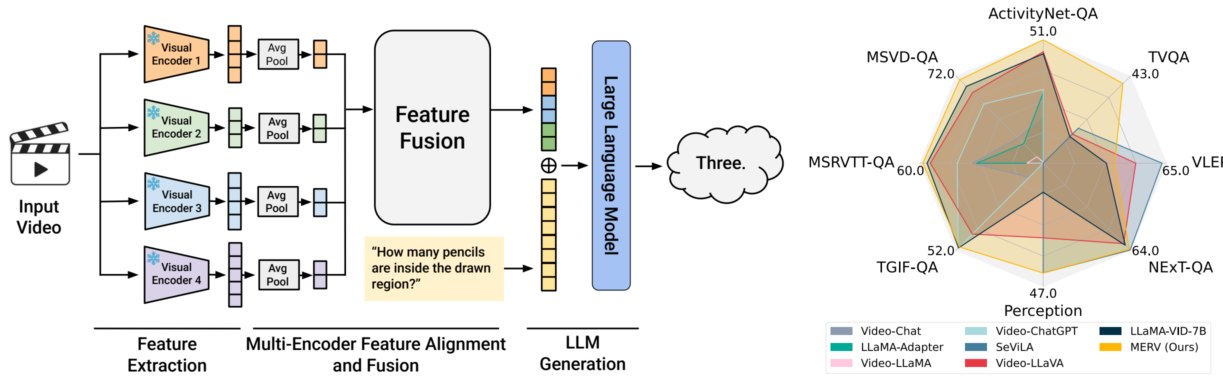


Figure 2. **MERV architecture and performance.** (**Left**) MERV proceeds in three main stages. First, we feed in our input video into each of visual encoders to get different representations. They are then spatio-temporally aligned before being fused by a cross-attentive mixer. The output is a visual embedding with an additive mix of information from all the encoders, which is combined with the text query to generate the result. **(Right)** We only compare MERV to the 7B prior models trained with comparable training data mixes. MERV is on-par or better than these single-encoder prior works, while deliver considerable gains over Video-LLaVA (Lin et al., 2024), from which MERV was initially adapted to form a Multi-Encoder Video Representation.

cases where only one of four different single-encoder models answers a given question correctly. While simple scene descriptions can be answered by image-level models, other questions require temporal and action-level comprehension, benefiting from features encoded with video models like ViViT (Arnab et al., 2021). Consequently, the reasoning capabilities of these VideoLLMs are directly limited by the inherent weaknesses of their respective pre-trained encoders. Therefore, employing multiple encoders could allow us to complement one encoder's weaknesses with another encoder's strengths. It also offers a cheap way to broaden the input distribution, leveraging different encoder's training mixtures. The wide adoption of the LLaVA paradigm is also indication that vision-language alignment is simple to achieve, even without language-aware vision models.

We propose *MERV*, a Multi-Encoder Representation of Videos, as a new method for integrating multiple visual encoders into a single VideoLLM using a cross-attentive encoder mixer for fusing representations. We introduce a spatio-temporally aligned representation for mixing the information from multiple types of visual encoders. Given the computational complexity of video tasks, we carefully experiment with optimization strategies and parallelizing the visual experts, allowing us to combine four distinct visual encoders with minimal computational overhead. Our frozen method outperforms all individual encoder methods, is up to 4.6% better than prior works (Lin et al., 2024) on video reasoning benchmarks, i.e., from 37.66% to 42.28% on TVQA (Lei et al., 2018), and on par with SeViLA (Yu et al., 2024b) on Perception Test (Pătrăucean et al., 2023), a challenging perception and reasoning diagnostic for video models. Finetuning the full model improves MERV past SeViLA (Yu et al., 2024b) by 2.2%, from 46.2% to 48.4%. Finally, we do a detailed qualitative study of our model's capabilities on the Something-Something v2 dataset (Goyal et al., 2017). We show that MERV can accurately capture both contrastive encoders' (Zhai et al., 2023; Zhu et al., 2024a) strengths on general vision-language understanding, as well as ViViT's (Arnab et al., 2021) specialty on temporally-sensitive tasks (e.g. distinguishing pushing left vs. right), without trading off performance between these specializations as single encoder models do. [1]

2. Related Works

VideoLLMs build upon the powerful reasoning capabilities of LLMs by utilizing them as language decoders to enable instruction-followed video understanding. Key advancements include VideoChat (Li et al., 2023c) and Video-LLaMA (Zhang et al., 2023) for chat-based video understanding, LLaMA-Adapter (Zhang et al., 2024b) for pre-alignment, Valley (Luo et al., 2023) with multilingual LLMs, InternVideo (Wang et al., 2022) with a dedicated video encoder training phase, and Video-ChatGPT (Maaz et al., 2024) combining video-adapted encoders with LLMs. GPT4Video (Wang et al., 2023) supports video understanding and generation, while MovieChat (Song et al., 2024) focuses on long video comprehension. Models like Chat-UniVi (Jin et al., 2024), LLaMA-VID (Li et al., 2024b), and VTM (Lee et al., 2024) optimize token usage for video representation. Other notable models include Vamos (Wang et al., 2024b), which flexibly uses visual embeddings, action labels, and video captions as input; VideoChat2 (Li et al., 2024a), developed through three-stage progressive training; Video-LLaVA (Lin et al., 2024), which aligns image and video representations before projecting them to the

[1]Our code and pretrained weights are available at `https://github.com/princetonvisualai/merv`.

LLM space; and VideoPrism (Zhao et al., 2024), which also further trains a video encoder through masked distillation. Specialized models like VTimeLLM (Huang et al., 2024) focus on fine-grained video moment understanding and time-bound reasoning, while models like Elysium (Wang et al., 2024a) and Merlin (Yu et al., 2024a) can predict object trajectories. SeViLA (Yu et al., 2024b) uses LLM for frame localizer of the video for multiple-choice tasks. Finally, recently LLaVA-Hound-DPO (Zhang et al., 2024a) explored using DPO and a higher quality training set for better instruction following. Distinct from these aforementioned works, our approach centers on utilizing a diverse array of visual and video encoders, which are RGB-based but coming from different visual backbone families, each with its own unique strengths, to significantly enhance the capabilities of the VideoLLM framework. By strategically using these specialized encoders, we aim to capture a broader spectrum of visual information, enriching VideoLLMs' understanding of videos.

Combining multiple encoders for multimodal LLMs is gaining attention. Eyes Wide Shut (Tong et al., 2024b) explored mixing DINOv2 and CLIP features for LLaVA, but their results signal that mixing features effectively requires investigation. Both Mipha (Zhu et al., 2024b) and Prismatic-VLMs (Karamcheti et al., 2024) found that image encoders like CLIP and SigLIP, which are trained using vision-language contrastive loss, surpass other image encoders such as ViT and DINOv2, with SigLIP showing further improvements over CLIP. SPHINX-X (Gao et al., 2024) and SPHINX (Lin et al., 2023) combines multiple image encoders by concatenating features along the channel dimension, while BRAVE (Kar et al., 2024) concatenates features from multiple encoders sequence-wise, followed by a QFormer with masked modeling. More recently, Cambrian-1 (Tong et al., 2024a) creates a multimodal LLM which spatially aligns their inputs across different resolutions. This also reproduces our finding of the importance of spatio-temporal alignment. There is also the popular body of research on multimodal LLMs using many modalities including image, video, audio and/or 3D (Chen et al., 2023; Han et al., 2023; Li et al., 2023a; Lyu et al., 2023; Panagopoulou et al., 2023; Su et al., 2023; Sun et al., 2023a; Zhang et al., 2023; Liu et al., 2024a; Han et al., 2024; Liu et al., 2024b; Jain et al., 2024). In contrast, this paper dives into the video-language domain, exploring combining multiple image and video encoders and exploiting their structural similarities. Our multi-encoder feature alignment and fusion are both performant and efficient in FLOPs, and results in an all-encompassing additive mixture of features which previous works could not achieve.

3. MERV: Multi-Encoder Representation of Videos

Our goal for MERV is to systematically build a video model that leverages multiple encoders with an LLM to process a video following the LLaVA/PrefixLM (Liu et al., 2023; 2018) paradigm (see Figure 2). Unlike previous works, our focus is not on combining multiple encoders of different *modalities* that make use of additional information (depth, audio, etc) (Bachmann et al., 2022; Zhu et al., 2024a), but instead focus on RGB-encoders trained on different datasets and objectives that offer *different* visual understanding. We extensively ablate three key aspects to make this possible: our selection of *multiple* encoders, i.e., which visual encoders and how many to use (Sec 3.1); how we *align* the spatio-temporal representations of each encoder to mix the information together, especially in an efficient manner (Sec 3.2); and our *implementation* efficiencies, from the parallel visual processing to the training recipes (Sec 3.3).

3.1. Multi-Encoder Feature Extraction

Our final architecture uses four distinct types of models: spatial experts, fine-grained temporal experts, image-language experts, and video-language experts. We found experimentally that our choice of four performed the best across all types of questions, and ablate our choices in Sec. 4.2. More details about these four encoders are in Appendix Table 4.

Spatial expert: DINOv2 (Oquab et al., 2023) is trained using unsupervised learning on local-to-global correspondences in image data. The resulting features have robust object part understanding, as well as semantic image understanding, but can suffer from poor language grounding.

Temporal expert: ViViT (Arnab et al., 2021) is trained using supervised learning on short videos. The architecture is designed for modeling the interactions between frames using spatial and temporal attention, which lets it capture longer temporal dependencies than pure image models can.

Image-Language contrastive expert: SigLIP (Zhai et al., 2023) is trained using sigmoid contrastive learning on image-text pairs. The model is designed to learn a joint embedding space for images and text, which makes it good at understanding vision-language associations. However, it can overlook the finer details of an image which are not well described by text in its training data.

Video-Language contrastive expert: Finally, our video-language expert is LanguageBind (Zhu et al., 2024a). Used by Video-LLaVA (Lin et al., 2024), LanguageBind is trained through joint multimodal learning between text and multiple modalities (videos, infrared, etc.,) and understands the relationship between video and text and their high-level semantics. We only use the video encoder of LanguageBind.

3.2. Spatio-Temporally Aligned Representations for Feature Fusion

Our input is a batch of text, image-text, or video-text queries. The visual part of the input, either images or videos, is passed through each of the visual encoders to extract the respective features. Here we describe the detailed care we took in pre-processing to prepare the features for alignment.

First, images are treated as videos with repeated frames, so assume all inputs are videos from here on out. A video is of shape $T \times H \times W$, where T is the number of frames and H, W are the height and width of the frames, and produce an output of shape $t_e \times h_e \times w_e$ for an encoder e. One obstacle with using different visual encoders is that each model outputs features with a different structure. For example, given an input of shape $16 \times 224 \times 224$, ViViT outputs a feature of shape $8 \times 14 \times 14$ whereas LanguageBind's features are of shape $16 \times 16 \times 16$. Image-based encoders will not change the temporal dimension, whereas ViViT downsamples the frames by a factor of 2.

For temporal alignment, as each encoder is flexible enough to handle varying input frames, we simply choose our input T for each encoder so that each output t_e is the same across all encoders, i.e. t.

Pre-fusion projection. Now we need to achieve spatial alignment among the features. Naïvely combining them will not work as they all have different spatial shapes. Using the full resolution features would also be prohibitively expensive. We design a pre-fusion projector to tackle both issues by aligning and compressing the features.

Suppose our feature from encoder e is $\mathbf{v}_e \in \mathbb{R}^{t \times h_e \times w_e \times d_e}$, where d_e is the dimension of encoder e, and assume the output spatial representations are square (i.e. $h_e = w_e$, but we keep notation for clarity). Our pre-fusion projector uses an adaptive 2D average pool $\mathcal{P}$ for each encoder to resize the spatial dimensions to the same $h \times w$ for all encoders, where $h < h_e$ and $w < w_e$. As t is the same across each $\mathbf{v}_e$, this *spatio-temporally aligns* the representations.

Finally, we need to connect the varying embedding dimensions d_e to a same dimensional space. We add a linear layer to project the features from dimension d_e to d, the LLM's dimension. In total, our pre-fusion projection is

$$\mathbf{x}_e := \mathcal{P}(\mathbf{v}_e) W_e \in \mathbb{R}^{\ell \times d} \quad \text{for } e \in \text{Encoders} \tag{1}$$

where $W_e \in \mathbb{R}^{d_e \times d}$ is each encoder's output linear layer, and $\ell = t \times h \times w$. This projector is lightweight, having only $d \times \sum_e d_e$ trainable parameters for dimension matching, making it easy to scale to an arbitrary number of visual encoders. For detailed ablations, see Section 4.2.1.

Feature fusion strategies. The final part of our pipeline is fusing the multi-encoder information together using cross-attention with learnable queries to additively mix the different representations together. The visual features determine the weights of the linear mixture, which we find sufficient for our task. We use a single randomly initialized query $\mathbf{Q} \in \mathbb{R}^{1 \times d}$, keys as $\overline{\mathbf{X}} = [\overline{\mathbf{x}}_1 \ \ldots \ \overline{\mathbf{x}}_N] \in \mathbb{R}^{N \times d}$, where $\overline{\mathbf{x}}_e \in \mathbb{R}^d$ is each encoder's features (after Eq. 1) averaged over the sequence dimension ℓ for a faster computation, and N the number of encoders, and values as $\mathbf{X} = [\mathbf{x}_1 \ \ldots \ \mathbf{x}_N] \in \mathbb{R}^{N \times \ell \times d}$. We calculate our final unified feature as

$$\mathbf{O} := \text{Softmax}\left(\frac{\mathbf{Q}\overline{\mathbf{X}}^\top}{\sqrt{d}}\right)\mathbf{X} \in \mathbb{R}^{\ell \times d}. \tag{2}$$

The final step is to concatenate the visual embedding and tokenized text together into the LLM. We use the base LLaMA-2 7B model (Touvron et al., 2023b), which we found performs better than the chat model. We test multiple alternate feature fusion strategies in Section 4.2.2.

3.3. Implementation Efficiencies

Parallelized visual encoding. At a first glance, using multiple encoders seems to be a large cost to pay when comparing the raw FLOPs and parameters. However, a key benefit of the LLaVA style architecture is that the entire feature extraction and projection pipeline can happen *in parallel*. To make this possible, we build on top of recent powerful advances in parallel processing for LLMs and use PyTorch's Fully Sharded Data Parallel (Zhao et al., 2023). As the video encoders themselves are much smaller than the LLM blocks, and their visual encoding processes are completed in roughly the same amount of time, most of the overhead in running four encoders is already covered by having just one encoder (*ref.* Figure 3). We provide some timing numbers in Section 4.2.3 and find that our step time is similar to that of the single-encoder methods.

Our code is built on top of the Prismatic VLM codebase (Karamcheti et al., 2024), which efficiently implements vision-language model (VLM) training. We add the ability to handle videos and an arbitrary number of visual encoders, along with many useful features for training. Our training is efficient for using multiple visual models, completing in under 24 hours using 8 L40-48GB GPUs, and down to 8 hours using 8 H100s. In contrast, the Video-LLaVA codebase runs the Stage-2 training in around 38 hours on the same L40 setup and could not easily support multiple encoders in our initial attempts.

MERV frozen and full. Many different recommendations for training LLaVA style models have been made since its inception. This is only made more complicated by the introduction of new datasets with every new VideoLLM architecture, making it difficult to properly determine the

Table 1. **Comparison of different multimodal LLMs on video reasoning benchmarks**. We employ ChatGPT to evaluate performance following Video-ChatGPT where applicable (version `gpt-3.5-turbo-0613`). * denotes our evaluation of using the author provided checkpoint. The first five datasets were used as evaluation test sets during development; the last three were held-out for our final evaluation.

Methods	MSVD-QA		MSRVTT-QA		TGIF-QA		Perception	ActivityNet-QA		NExT-QA	VLEP	TVQA
	Acc	Score	Acc	Score	Acc	Score	Acc	Acc	Score	Acc	Acc	Acc
Alternative data mixes										-		
Video-Chat (Li et al., 2023c)	56.3	2.8	45.0	2.5	-	-	-	26.5	2.2	-	-	-
LLaMA-Adapter (Zhang et al., 2024b)	54.9	3.1	43.8	2.7	-	-	-	34.2	2.7	-	-	-
Video-LLaMA (Zhang et al., 2023)	51.6	2.5	29.6	1.8	-	-	-	12.4	1.1	-	-	-
Video-ChatGPT (Maaz et al., 2024)	64.9	3.3	49.3	2.8	-	-	-	35.2	2.7	-	-	-
SeViLA (Yu et al., 2024b)	-	-	-	-	-	-	46.2	-	-	**63.6**	**64.4**	38.2
LLaMA-VID-7B* (Li et al., 2024b)	69.30	3.74	57.84	3.24	51.31	3.26	41.64	46.45	3.22	60.61	57.65	37.43
LLaMA-VID-13B* (Li et al., 2024b)	70.25	3.77	58.58	3.26	51.26	3.26	41.54	46.79	3.23	60.03	61.98	41.33
Same data mixes										-		
Video-LLaVA* (Lin et al., 2024)	67.74	3.69	56.90	3.18	47.99	3.17	44.22	47.08	3.27	59.61	61.21	37.66
MERV (frozen) / MERV	**70.97**	3.76	**59.03**	**3.25**	51.1	3.26	46.21	**50.87**	**3.34**	63.09	58.66	**42.28**
Gains to Video-LLaVA*	+3.23	+.07	+2.13	+.07	+3.11	+.09	+1.99	+3.79	+.07	+3.48	-2.55	+4.62
MERV (full)	70.48	**3.79**	57.25	3.24	**51.39**	**3.28**	**48.41**	49.93	3.33	61.36	60.07	39.42
Gains to Video-LLaVA*	+2.74	+.10	+0.35	+.06	+3.40	+.11	+4.19	+2.85	+.06	+1.75	-1.14	+1.76

best recipe for one's own setup. We intentionally fix our dataset to be the same as Video-LLaVA's so we can isolate the impacts of the training setup, from which we find two viable settings: *MERV (frozen)* and *MERV (full)*. The original Video-LLaVA's recipe has two stages: a Stage 1 pre-training on captioning data to align only the connector between the pre-trained vision encoder and LLM, and a Stage 2 instruction tuning on both connector and the LLM.

- *MERV (frozen)*, which performs only the Stage 2 instruction tuning inspired by Karamcheti et al. (2024); it achieves similar results to the original Video-LLaVA recipe in only 43% of the time;
- *MERV (full)*, which undergoes both of the Stage 1 captioning pre-training and the Stage 2 instruction tuning, but also unfreezes the LLM during Stage 1 for a slight improvement on a few benchmarks.

As MERV (frozen) is faster to train with similar performance, we adopt that recipe by default for analysis, and interchangeably use *MERV* to refer to it for simplicity from here on out. Detailed analysis is provided in Section 4.2.3.

4. Experimental Results

In this section, we show that our method outperforms prior works across standard video-language benchmarks before moving onto an in-depth analysis of our method and its specializations in the next section.

Datasets and training procedure details. For fair comparison, our data mix is the same as Video-LLaVA (Lin et al., 2024). The Stage 1 data is single-turn concise captioning, with 558k (image, text) pairs from LAION filtered by LLaVA (Liu et al., 2023) and 702k (video, text) pairs from Valley (Luo et al., 2023). The Stage 2 data is multi-turn conversations, detailed captioning and reasoning, with 665k (image, text) pairs from LLaVA and 100k (video, text) instructions from Video-ChatGPT (Maaz et al., 2024).

All the preprocessing, including frame extraction, adheres to the original method that each encoder is trained with. We extract 16 uniformly sampled frames from each video, except for ViViT which extracts 32 frames by default but produces a 16-frame output feature.

For MERV (frozen), we train on only Stage 2 data for 1 epoch with a learning rate of 2×10^{-5} and a batch size of 128 with gradient accumulation. For MERV (full), we first train on Stage 1 data with a learning rate of 1×10^{-4} and the projectors, feature fusion, and LLM unfrozen with similar settings. Both recipes use an initial warmup ratio of 0.03 and a cosine schedule.

Evaluation. We evaluate our model on a comprehensive suite of video understanding benchmarks, including the open-ended MSVD-QA (Xu et al., 2017), MSRVTT-QA (Xu et al., 2017), TGIF (Jang et al., 2017), and ActivityNet-QA (Yu et al., 2019), as well as the multiple-choice benchmarks NExT-QA (Xiao et al., 2021), VLEP (Lei et al., 2020), TVQA (Lei et al., 2018), and Perception Test (Pătrăucean et al., 2023). We emphasize that NExT-QA, VLEP, and TVQA datasets are **held-out** datasets that we did not use during our experiments, and only evaluated once after all the design is completed. We report both accuracy and score following the Video-ChatGPT evaluation protocol where applicable, and all evaluations are done zero-shot without any dataset-specific fine-tuning. Results using GPT-3.5-turbo for evaluation are done with the June 13th, 2023 cutoff date.

4.1. Comparison to State of the Art

Table 1 tabulates the performance of MERV (frozen) and (full). We compare our model to the existing works, including Video-LLaVA (Lin et al., 2024) that share our train-

Table 2. **Ablating design choices.** We highlight our defaults in orange and **bold** the best results. Average accuracy is on MSVD, MSRVTT, TGIF, and Perception Test. Full metric results are in the Appendix.

(a) **Pre-fusion projectors.** * is 16 frames instead of 8. Top two rows are projector-free baselines.

Projector	Avg Acc	Params	FLOPs
257 tok	54.76	-	-
class tok	52.05	-	-
2D Avg	54.96	0	2.1M
2D Avg*	**55.86**	0	4.2M
2D Attn	52.12	12.7M	9.7G
2D Conv	54.23	237M	241G
3D Avg*	55.09	0	4.2M
3D Conv	55.42	113M	232G

(b) **Pre-fusion output token.** We ablate the optimal token size per frame for the pre-fusion projector.

Tkns	MSVD	MSRVTT	TGIF
1	61.94	54.64	41.41
4	64.47	55.72	45.32
16	67.23	56.44	47.75
64	**69.08**	**58.00**	**50.01**
100	68.38	57.47	48.78
144	68.65	57.73	48.81
256	68.46	57.72	48.66

(c) **Feature fusion strategy.** Cross-Attn additive mixing is the best overall among all the strategies on accuracy, for its FLOPs.

Strategy	Avg Acc	FLOPs
Cross-Attn	**56.83**	17.19 T
Concat (Seq.)	54.45	43.09 T
Concat (Ch.)	56.64	16.29 T
Learnable W	55.01	16.24 T
25% - Mixed	54.19	16.39 T

ing data mixture, and other VideoLLMs (Li et al., 2023c; 2024b; Maaz et al., 2024; Yu et al., 2024b; Zhang et al., 2023; 2024b). We find that our method, generating video representations using multiple visual encoders that specialize in different skills of video understanding, outperforms Video-LLaVA across nearly all of the benchmarks, with a 4.1% gain on Perception Test, a 3.7% gain on ActivityNet and a 4.6% gain on TVQA. Both of our methods perform better overall than Video-LLaVA, even when using less data with just Stage 2 as shown by the MERV (frozen) numbers. While MERV (full) is not a strict improvement to MERV, it still improves on some difficult benchmarks (e.g., Perception Test) with its additional video-language alignment. We believe that this makes MERV (full) generalize better in unseen settings outside of these testing benchmarks, and recommend using this recipe when possible. Compared to LLaMA-VID-7B, which uses a different training mix, we are better in nearly all benchmarks, up to around 4.5% across Perception Test, ActivityNet, and TVQA. Moreover, MERV (full) outperforms SeViLA on the Perception Test zero-shot with 48.4%, compared to 46.2%. Overall, our design shows a significant improvement over Video-LLaVA and prior methods as a whole.

4.2. Ablations

In this section, we justify our design choices for the projectors, feature fusion strategies, and training recipes. Our ablations are done with the MERV (frozen) recipe.

4.2.1. Pre-Fusion Projectors

The first module we investigate is our projectors, which serve to connect each encoder from its pre-trained embedding space to a common embedding space. We test two types of projectors: image-level, which operate on frames independently, and video-level, which aggregate information across frames. Projector details are provided in Section A.3 in the Appendix. We report average performance across our development sets of MSVD, MSRVTT, TGIF, and Perception Test.

Pre-fusion projector. Table 2a tabulates our projector ablation on average accuracy, parameter count, and FLOPs, with the default settings of LanguageBind as the single vision encoder and an 8 frame 64 token projection output. We find that 2D average pooling is the best, with 55.86% average accuracy, even better than using no projector on the full 257 token representation (as in Video-LLaVA (Lin et al., 2024)), with 54.76% average accuracy. It also has no trainable parameters and the fewest FLOPs. The projection serves as a form of feature selection, allowing the LLM to efficiently reason only about the most relevant information.

One result worth noting is the poor performance (52.12%, 3.7% lower) of the attentive resampler, a popular projector choice. The attentive resampler is less responsive to the input spatio-temporal structure compared to other projectors, which leads it to being a weaker projector for us. However, it would likely be better with higher quality data from which the transformer can learn. Increasing the frame resolution from 8 to 16 was also a large improvement, showing that increasing temporal resolution is still important. This highlights the importance of aligning representations with their spatial and temporal structure, *especially* for video models, which extract many more frames of visual information.

Projector token length. Similarly, we ablate the output token length of the projector. Table 2b tabulates the performance of producing 1 to 256 tokens for a 2D average pooling projector with 16 frames as input. We see that the performance peaks at 64 tokens with 69.08% MSVD accuracy, with worse performance for longer token lengths (max. 68.65%). This balances the number of tokens used for condensing the visual embedding while also minimizing the extra processing needed by the LLM.

4.2.2. Feature Fusion Strategies

Next, we test different strategies for fusing the information from all of the features, with detailed breakdowns in Table 2c. First, we evaluate two popular concatenation methods, in either the token sequence dimension, or the channel dimension followed by an MLP for matching the LLM dimension. While sequence-wise concatenation is widely used in multimodal LLMs (Tong et al., 2024b), our method outperforms it while using significantly less computation, with a 56.8% average accuracy compared to 54.4%, while also using 2.5× fewer FLOPs. Concatenation channel-wise reaches a similar performance of 56.6% and a lightweight cost. However, our cross-attention shows better average performance, better or on par on 3 out of 4 benchmarks (*ref.* Appendix Table 6c), with the additional benefit of having accessible encoder weightings for analysis, so we stick with cross-attention for our final design. We also try different methods of additive mixing as an ablation. The last two rows of Table 2c show the performance when either learning the additive weights as a single scalar or by fixing the weights to be 0.25 for each of the 4 encoders. We see that using cross attention outperforms both methods by 1.8% and 2.6%, as our feature fusion module can dynamically generate better fused embeddings based on the visual input.

4.2.3. Training Recipes

Finally, we also compare different training recipes based on the literature and our own expertise. Apart from training recipes mentioned in Section 3.3, many recent works have attempted some combination of other strategies, such as unfreezing the vision encoders in Diao et al. (2024), which usually requires more training data.

We systematically map out this landscape, fixing our dataset to be the same as Video-LLaVA's. Unlike them, we found that the Stage 1 phase did not help much when training only the projectors and feature fusion (*ref.* Appendix Table 5), with roughly the same average accuracy. Stage 2 instruction tuning alone leads to similar results in 43% of the total time, so we adopt this recipe for efficiency and refer to it as *MERV (frozen)*. This recipe is still unsatisfying as it leaves a large amount of data, approximately 1.3M vision-text pairs, unused for training. In our empirical observations, we found that the resulting video-language alignment was suboptimal. The distributions of language used in video training datasets and benchmarks sparsely overlap based on their sentence embeddings, which could be impacting our ability to generalize zero-shot on downstream benchmarks. We address this by *unfreezing* the LLM during Stage 1 to better learn this alignment, improving performance on a few key benchmarks, especially Perception Test, by up to 2.2%. We call this recipe *MERV (full)*.

As another ablation, we train MERV on a single stage comprised of the Stage 1 and Stage 2 data mixed together (*ref.* bottom of Appendix Table 5). Surprisingly, this does worse than the explicit two stage training recipe. We attribute this to the explicit types of data in each stage being a form curriculum learning, showing that these stages are still important for optimal performance.

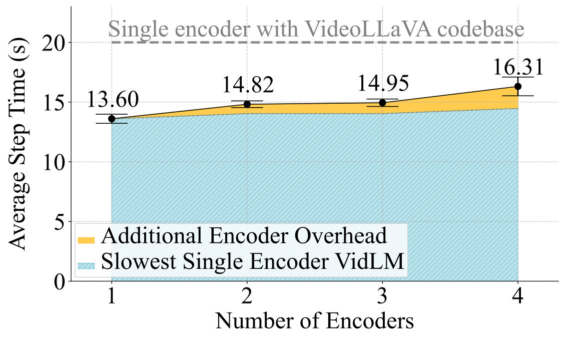


Figure 3. **Extra encoders incur minimal step time overhead**. We add encoders in the order of DINOv2, LanguageBind, SigLIP, ViViT, plotted alongside the slowest single encoder in each group.

Finally, we provide evidence for the efficiency of our method (*ref.* Figure 3). We use the default FSDP sharding strategy PyTorch provides; it is not currently possible to specify explicit plans for which modules go where (but may be possible as FSDP matures). However, even with this basic strategy, our method is dominated by the slowest single encoder present, incurring very little additional overhead from extra encoders due to this parallelization, making it cost-efficient to scale up in the number of encoders.

4.3. Ensemble Composition

The original motivation of our work was to choose encoders with complementary visual knowledge to form a comprehensive representation for our final model. The key questions are 1) do we benefit by using more than one encoder, and 2) do we need all four encoders, i.e. does each one meaningfully contribute to the final performance?

Can we make use of more encoders? The conventional wisdom is to use a single encoder, typically a contrastively trained vision-language model like CLIP, SigLIP, or LanguageBind (Radford et al., 2021; Zhai et al., 2023; Zhu et al., 2024a), in a VideoLLM. In Figure 4a (●) , we show the four single encoder models corresponding to each of our chosen encoders using their full embeddings. They not only all perform worse than MERV but also use more FLOPs, as without our pre-fusion projectors, their sequence lengths are at least 4× ours.

Are each of the encoders contributing? To affirm that this set of four encoders is actually beneficial for improving understanding, we train three-encoder VideoLLMs under the same strategy, but removing a different encoder each time. Each of these models does worse based on the strength of the encoder removed, meaning that MERV is using their knowledge (*ref.* Fig. 4a, ▲). The minor drop in FLOPs illustrates how most of the computation is still dominated by the LLM, not the vision encoders.

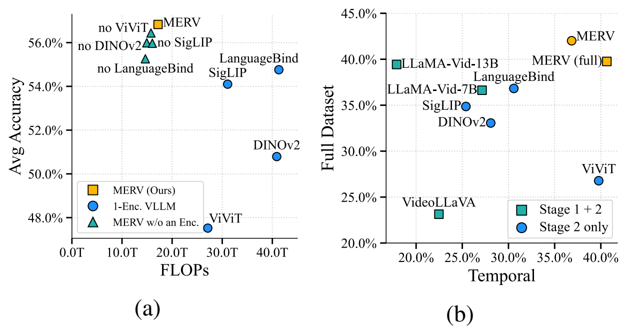


Figure 4. **Analysis plots supporting our design of multiple encoders, from their accuracy to their skill specializations.**
(a) Visual Encoder Subsets. MERV outperforms single-encoder VideoLLMs (●), with our feature projectors unlocking more computational efficiency. Removing any encoder also reduces MERV performance (▲). Average accuracy is across MSVD, MSRVTT, TGIF, and Perception Test. **(b) SSv2-MCQ and Temporal.** Temporal denotes performance on 12 selected classes where actions are indistinguishable if played in reverse (*ref.* Sec. 5.3). Full results are in the Appendix Tables 7, 8, 10.

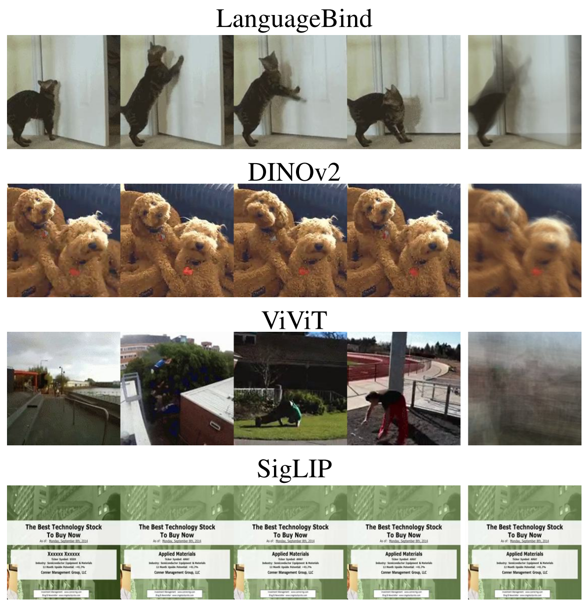


Figure 5. **Videos that give the highest attention weight for each of the encoders.** The right-most column shows the average frame of the video. For more examples, see Figure 11 in the Appendix.

5. Analysis

5.1. Cross Attention Weights Activate on Corresponding Videos

We first understand our method by looking at the cross attention weights on our 4 benchmark datasets (MSRVTT, TGIF, MSVD, and Perception Test), and visualizing the videos which have the highest attention weight for each of the encoders in Figure 5. This lets us see what types of videos activate each encoder the most. As expected, ViViT attention weights are highest on videos with large motion, as ViViT has strong temporal understanding. Meanwhile, SigLIP is utilized for videos that have textual data in the video, likely due to being vision-language contrastively trained, especially with textual data during training. DINOv2 and LanguageBind are both preferred by videos with static scenes, but LanguageBind is preferred for videos with foreground motion.

5.2. MERV Can Capture Visual Skills of Different Encoders

Next, we ask if our model effectively captures knowledge from its encoders. We first answer through our previous open-ended QA benchmarks. To assess the performance across different visual tasks, we create "pseudo"-skill categories by looking at the first word of the question sentence, which are often WH-words. They can be viewed as a proxy of skills required to solve the task. For example, `Where` requires spatial understanding and `When` requires temporal understanding. Figure 6 shows the relative performance of different visual encoders. While the contrastive models generally dominate each category, no single encoder performs best in all tasks. LanguageBind, for example, performs the best in TGIF-`What` with 46.23%, while DINOv2 performs on par with the best in MSVD-`Who` with 82.12%. Our method combines different encoders into an unified representation and consistently matches or improves the best-performing single-encoder model. Raw numbers are in Table 9 in the Appendix.

5.3. MERV Can Intuit Motion and General Understanding Simultaneously

We take another angle to quantifying how well our model learns from each of its individual encoders by looking back to classic video action recognition datasets. They are commonly used for general action understanding, but they can also be used to evaluate finer-grained capabilities. We are most interested in both general understanding and distinguishing actions which are temporally ambiguous, i.e., indistinguishable when reversed in time, such as *Pulling [something] from left to right* and *Pulling [something] from right to left*. This offers a fair analysis of both general and finer-grained video understanding.

We turn to the Something-Something v2 (Goyal et al., 2017) (SSv2) dataset, where the goal is to classify an input video into one of 174 classes, e.g., *Pulling [something] from left to right*. This allows us to analyze our model's understanding of temporal-spatial interaction with minimal distractions from scene understanding and real-world semantics. However, evaluating SSv2 as a zero-shot open-ended task is difficult with a long tail of specific categories. Thus we reformat the dataset into 5-way multiple-choice questions

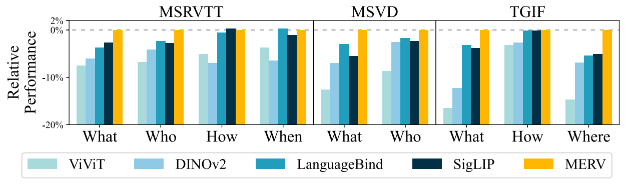


Figure 6. **Single encoder vs. MERV on different types of video tasks**. We plot the relative performance of VideoLLMs with different visual encoders. While each single encoder has its strength in different tasks, our method shows better performance than all the other single encoders in almost every task. We only plot tasks with more than 500 samples. See Appendix for details.

(MCQ) and fix the prompt to be "How is the object in the video being interacted with?". Incorrect choices were randomly sampled from the other 173 classes. We call this benchmark "SSv2 - MCQ" to distinguish it from the original classification task. Additionally, we selected 12 classes *a priori* from SSv2, where the action is indistinguishable if reversed in time, e.g., *Pulling [something] from left to right* and *Pulling [something] from right to left* to form a subset that can test the model's ability in temporal understanding.

Figure 4b plots the performance of MERV and single-encoder models on this temporal subset (x-axis) against the full dataset (y-axis). We see that the ViViT single-encoder model, which often falls short in other video QA benchmarks, surprisingly performs better than other encoders at 39.77% on temporal subset, which is 9.19% higher than the next closest model LanguageBind. However for the full SSv2 - MCQ, ViViT suffers with a worse performance of 26.78%, as ViViT's strength is on temporal understanding despite lacking in vision-language understanding. Contrastive encoders have the upper-hand in most other classes.

We believe that the architecture, datasets, and objective of each model causes these difference. ViViT processes spatial-temporal tubelets for embeddings, leading to better temporal understanding despite only being pre-trained on Kinetics-400 classification. SigLIP uses image-based ViT with no temporal layer has limited temporal understanding, but has a greater knowledge due to its larger training set and contrastive objective. MERV, at 42%, shows better performance compared to all these single-encoder models via leveraging strength of all the individual encoders. MERV (full) performs better than both VideoLLaVA (Lin et al., 2024) and the 7B and 13B variants of LLaMA-Vid (Li et al., 2024b).

Finally, we also plot the performances of six SSv2 classes where the performance difference between ViViT and SigLIP is largest in Figure 7. We see that actions which cannot be inferred from a single frame are the ones that ViViT performs better, e.g., *Pushing [something] from left to right* is indistinguishable from *Pushing [something] from right to left* if temporal information is omitted. Meanwhile, SigLIP performs better for classes where understanding the

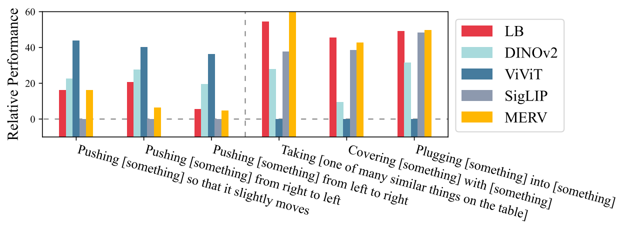


Figure 7. **Single-Encoder Performance Difference in Something-Something v2 - MCQ**. ViViT shows better performance on tasks where temporal understanding is crucial, while LanguageBind (LB) and SigLIP show better performance where task can be solved from single-frame understanding.

semantics of the scene can hint the action that is happening, e.g., if a frame where a blanket is dropped on an object is given, one can easily expect *Covering [something] with [something]* without watching the full video. See Appendix Figure 9 for sample videos of the these classes.

6. Conclusion

Previous VideoLLMs have been limited to relying on a single visual model for feature extraction, which leads to limited understanding capabilities of vastly different video tasks. In our work, we break this paradigm and explore various fusion strategies for combining information from multiple visual experts to generate a representation that can leverage the capabilities of different video encoders. We find that our multi-encoder feature fusion is able to outperform comparable methods by up to 4.62% on video reasoning benchmarks. We show that the method can obtain better performance than the best-performing single-encoder model with minimal computational overhead. Finally, we quantitatively and qualitatively observe the skill specializations our model learns on an MCQ format of Something-Something v2, which confirms both that encoders can be specialized and that our model captures both axes of knowledge. Our paper proposes some initial steps in rethinking how we approach the use of multiple encoders. We hope that this inspires others to also consider this problem as another direction for scaling and improving their VideoLLMs.

Acknowledgements

This work is supported by the National Science Foundation under Grant No. 2107048 and the Princeton First Year Fellowship to TZ. We also thank Allison Chen and William Yang for detailed comments and feedback.

Impact Statement

Our work aims to facilitate video understanding, which can lead to positive social impacts such as a video captioning model for low-vision users, automatic detection of medical emergencies, or better self-driving cars. It can also lead to negative social impacts like easy surveillance by the authorities, and human-like internet bots being used for scamming purposes. We follow the same safeguards implemented by the original authors of the datasets, the visual models, and the LLM models. We have not put any additional safeguards ourselves.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 2022.

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. Vivit: A video vision transformer. In *ICCV*, 2021.

Bachmann, R., Mizrahi, D., Atanov, A., and Zamir, A. Multimae: Multi-modal multi-task masked autoencoders. In *ECCV*, 2022.

Bain, M., Nagrani, A., Varol, G., and Zisserman, A. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021.

Bardes, A., Garrido, Q., Ponce, J., Rabbat, M., LeCun, Y., Assran, M., and Ballas, N. Revisiting feature prediction for learning visual representations from video. *TMLR*, 2024.

Cha, J., Kang, W., Mun, J., and Roh, B. Honeybee: Locality-enhanced projector for multimodal llm. In *CVPR*, 2024.

Chen, F., Han, M., Zhao, H., Zhang, Q., Shi, J., Xu, S., and Xu, B. X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages. *arXiv*, 2023.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 2023.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

contributors, S. Stablelm: Stability ai language models, 2023. URL https://github.com/Stability-AI/StableLM.

Diao, H., Cui, Y., Li, X., Wang, Y., Lu, H., and Wang, X. Unveiling encoder-free vision-language models. *NeurIPS*, 2024.

Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., and Cao, Y. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023.

Gao, P., Zhang, R., Liu, C., Qiu, L., Huang, S., Lin, W., Zhao, S., Geng, S., Lin, Z., Jin, P., et al. Sphinx-x: Scaling data and parameters for a family of multi-modal large language models. *ICML*, 2024.

Goyal, R., Kahou, S. E., Michalski, V., Materzyńska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., and Memisevic, R. The ”something something” video database for learning and evaluating visual common sense. In *arXiv*, 2017.

Han, J., Zhang, R., Shao, W., Gao, P., Xu, P., Xiao, H., Zhang, K., Liu, C., Wen, S., Guo, Z., et al. Imagebind-llm: Multi-modality instruction tuning. *arXiv*, 2023.

Han, J., Gong, K., Zhang, Y., Wang, J., Zhang, K., Lin, D., Qiao, Y., Gao, P., and Yue, X. Onellm: One framework to align all modalities with language. In *CVPR*, 2024.

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

Jain, J., Yang, J., and Shi, H. VCoder: Versatile Vision Encoders for Multimodal Large Language Models. In *CVPR*, 2024.

Jang, Y., Song, Y., Yu, Y., Kim, Y., and Kim, G. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017.

Jin, P., Takanobu, R., Zhang, C., Cao, X., and Yuan, L. Chat-univi: Unified visual representation empowers large language models with image and video understanding. In *CVPR*, 2024.

Kar, O. F., Tonioni, A., Poklukar, P., Kulshrestha, A., Zamir, A., and Tombari, F. Brave: Broadening the visual encoding of vision-language models. In *ECCV*, 2024.

Karamcheti, S., Nair, S., Balakrishna, A., Liang, P., Kollar, T., and Sadigh, D. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *ICML*, 2024.

Lee, S.-H., Wang, J., Zhang, Z., Fan, D., and Li, X. Video token merging for long video understanding. *Advances in Neural Information Processing Systems*, 37:13851–13871, 2024.

Lei, J., Yu, L., Bansal, M., and Berg, T. L. Tvqa: Localized, compositional video question answering. In *EMNLP*, 2018.

Lei, J., Yu, L., Berg, T. L., and Bansal, M. What is more likely to happen next? video-and-language future event prediction. In *EMNLP*, 2020.

Li, B., Zhang, Y., Chen, L., Wang, J., Yang, J., and Liu, Z. Otter: A multi-modal model with in-context instruction tuning. *arXiv*, 2023a.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023b.

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

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

Li, Y., Wang, C., and Jia, J. Llama-vid: An image is worth 2 tokens in large language models. In *ECCV*, 2024b.

Lin, B., Zhu, B., Ye, Y., Ning, M., Jin, P., and Yuan, L. Video-llava: Learning united visual representation by alignment before projection. In *EMNLP*, 2024.

Lin, Z., Liu, C., Zhang, R., Gao, P., Qiu, L., Xiao, H., Qiu, H., Lin, C., Shao, W., Chen, K., et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv*, 2023.

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

Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., and Lee, Y. J. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024a. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. Generating wikipedia by summarizing long sequences. In *ICLR*, 2018.

Liu, S., Fan, L., Johns, E., Yu, Z., Xiao, C., and Anandkumar, A. Prismer: A vision-language model with multi-task experts. *TMLR*, 2024b.

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

Lyu, C., Wu, M., Wang, L., Huang, X., Liu, B., Du, Z., Shi, S., and Tu, Z. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *arXiv*, 2023.

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

McKinzie, B., Gan, Z., Fauconnier, J.-P., Dodge, S., Zhang, B., Dufter, P., Shah, D., Du, X., Peng, F., Weers, F., Belyi, A., Zhang, H., Singh, K., Kang, D., Jain, A., Hè, H., Schwarzer, M., Gunter, T., Kong, X., Zhang, A., Wang, J., Wang, C., Du, N., Lei, T., Wiseman, S., Yin, G., Lee, M., Wang, Z., Pang, R., Grasch, P., Toshev, A., and Yang, Y. Mm1: Methods, analysis & insights from multimodal llm pre-training. In *ECCV*, 2024.

OpenAI, R. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5), 2023.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.-Y., Xu, H., Sharma, V., Li, S.-W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., and Bojanowski, P. Dinov2: Learning robust visual features without supervision. In *TMLR*, 2023.

Panagopoulou, A., Xue, L., Yu, N., Li, J., Li, D., Joty, S., Xu, R., Savarese, S., Xiong, C., and Niebles, J. C. X-instructblip: A framework for aligning x-modal instruction-aware representations to llms and emergent cross-modal reasoning. *arXiv*, 2023.

Pătrăucean, V., Smaira, L., Gupta, A., Continente, A. R., Markeeva, L., Banarse, D., Koppula, S., Heyward, J., Malinowski, M., Yang, Y., Doersch, C., Matejovicova, T., Sulsky, Y., Miech, A., Frechette, A., Klimczak, H., Koster, R., Zhang, J., Winkler, S., Aytar, Y., Osindero, S., Damen, D., Zisserman, A., and Carreira, J. Perception test: A diagnostic benchmark for multimodal video models. In *NeurIPS*, 2023. URL https://openreview.net/forum?id=HYEGXFnPoq.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

Ryali, C., Hu, Y.-T., Bolya, D., Wei, C., Fan, H., Huang, P.-Y., Aggarwal, V., Chowdhury, A., Poursaeed, O., Hoffman, J., Malik, J., Li, Y., and Feichtenhofer, C. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *ICML*, volume 202 of *ICML'23*, pp. 29441–29454, Honolulu, Hawaii, USA, July 2023. JMLR.org.

Song, E., Chai, W., Wang, G., Zhang, Y., Zhou, H., Wu, F., Guo, X., Ye, T., Lu, Y., Hwang, J.-N., et al. Moviechat: From dense token to sparse memory for long video understanding. In *CVPR*, 2024.

Su, Y., Lan, T., Li, H., Xu, J., Wang, Y., and Cai, D. Pandagpt: One model to instruction-follow them all. *arXiv*, 2023.

Sun, G., Yu, W., Tang, C., Chen, X., Tan, T., Li, W., Lu, L., Ma, Z., and Zhang, C. Fine-grained audio-visual joint representations for multimodal large language models. *arXiv*, 2023a.

Sun, Q., Fang, Y., Wu, L., Wang, X., and Cao, Y. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023b.

Tong, S., Brown, E., Wu, P., Woo, S., Middepogu, M., Akula, S. C., Yang, J., Yang, S., Iyer, A., Pan, X., et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv*, 2024a.

Tong, S., Liu, Z., Zhai, Y., Ma, Y., LeCun, Y., and Xie, S. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, 2024b.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. LLaMA: Open and Efficient Foundation Language Models, February 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2023b.

Wang, H., Ye, Y., Wang, Y., Nie, Y., and Huang, C. Elysium: Exploring object-level perception in videos via mllm. In *ECCV*, 2024a.

Wang, S., Zhao, Q., Do, M. Q., Agarwal, N., Lee, K., and Sun, C. Vamos: Versatile action models for video understanding. In *ECCV*, 2024b.

Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv*, 2022.

Wang, Z., Wang, L., Zhao, Z., Wu, M., Lyu, C., Li, H., Cai, D., Zhou, L., Shi, S., and Tu, Z. Gpt4video: A unified multimodal large language model for lnstruction-followed understanding and safety-aware generation. In *CoRR*, 2023.

Xiao, J., Shang, X., Yao, A., and Chua, T.-S. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021.

Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., and Zhuang, Y. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017.

Xue, H., Sun, Y., Liu, B., Fu, J., Song, R., Li, H., and Luo, J. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. In *ICLR*, 2023.

Yu, E., Zhao, L., Wei, Y., Yang, J., Wu, D., Kong, L., Wei, H., Wang, T., Ge, Z., Zhang, X., et al. Merlin: Empowering multimodal llms with foresight minds. In *ECCV*, 2024a.

Yu, S., Cho, J., Yadav, P., and Bansal, M. Self-chained image-language model for video localization and question answering. *NeurIPS*, 2024b.

Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., and Tao, D. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, 2019.

Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *ICCV*, 2023.

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

Zhang, R., Gui, L., Sun, Z., Feng, Y., Xu, K., Zhang, Y., Fu, D., Li, C., Hauptmann, A., Bisk, Y., and Yang, Y. Direct preference optimization of video large multimodal models from language model reward, 2024a.

Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., and Qiao, Y. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. In *ICLR*, 2024b.

Zhao, L., Gundavarapu, N. B., Yuan, L., Zhou, H., Yan, S., Sun, J. J., Friedman, L., Qian, R., Weyand, T., Zhao, Y., et al. Videoprism: A foundational visual encoder for video understanding. In *ICML*, 2024.

Zhao, Y., Gu, A., Varma, R., Luo, L., Huang, C.-C., Xu, M., Wright, L., Shojanazeri, H., Ott, M., Shleifer, S., Desmaison, A., Balioglu, C., Damania, P., Nguyen, B., Chauhan, G., Hao, Y., Mathews, A., and Li, S. Pytorch fsdp: Experiences on scaling fully sharded data parallel, 2023.

Zhu, B., Lin, B., Ning, M., Yan, Y., Cui, J., Wang, H., Pang, Y., Jiang, W., Zhang, J., Li, Z., et al. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. In *ICLR*, 2024a.

Zhu, M., Zhu, Y., Liu, X., Liu, N., Xu, Z., Shen, C., Peng, Y., Ou, Z., Feng, F., and Tang, J. A comprehensive overhaul of multimodal assistant with small language models. In *AAAI*, 2024b.

A. Appendix / Supplemental Material

A.1. Limitations

Our work is based on the LLM, LLaMA-2 7B model (Touvron et al., 2023b), and as with many other VideoLLM models, the performance of our method is hugely dependent on the capabilities of the LLM model, and better-performing models often demand significantly more computation. MERV requires an LLM and running multiple encoders, which can be computationally intensive and can lead to out-of-memory errors in resource-limited settings (while our efficient implementation alleviates the issue). While FSDP (Zhao et al., 2023) allows us to easily and effectively train larger models across multiple GPUs than would otherwise be possible, its generality also makes it difficult for us to design tailored sharding strategies that would maximize the performance of our model. However, with future improvements to data parallelism, our model can still benefit greatly and run even more efficiently. Also, despite the improved speeds, there is still an upper bound for what constitutes a *reasonable* training time that still allows us to test many of our design assumptions, which limits the scale and number of experiments we can run. While we show that our method can successfully leverage information from different visual encoders, nevertheless if the encoders themselves are limited in video understanding capability, MERV cannot fully compensate for that.

A.2. Model Details

A.2.1. Baseline Encoder and LLM Details

Table 3. **Visual Encoder and LLM Information**.

Model	Visual Encoder	LLM
Video-Chat (Li et al., 2023c)	ViT-G (EVA-CLIP) (Sun et al., 2023b)	StableVicuna (contributors, 2023)
LLaMA-Adapter (Zhang et al., 2024b)	CLIP (Radford et al., 2021)	LLaMA-1 7B (Touvron et al., 2023a)
Video-LLaMA (Zhang et al., 2023)	ViT-G (EVA-CLIP) (Sun et al., 2023b) + BLIP-2 Q-Former (Li et al., 2023b)	Vicuna-7B v0 (Chiang et al., 2023)
Video-ChatGPT (Maaz et al., 2024)	CLIP (Radford et al., 2021)	Vicuna-7B v1.1 (Chiang et al., 2023)
SeViLA (Yu et al., 2024b)	ViT-G (EVA-CLIP) (Sun et al., 2023b) + BLIP-2 Q-Former (Li et al., 2023b)	FlanT5-XL (3B) (Chung et al., 2024)
LLaMA-VID-7B (Li et al., 2024b)	EVA-G (Fang et al., 2023)	Vicuna-7B v1.5 (Chiang et al., 2023)
LLaMA-VID-13B (Li et al., 2024b)	EVA-G (Fang et al., 2023)	Vicuna-13B v1.5 (Chiang et al., 2023)
Video-LLaVA* (Lin et al., 2024)	LanguageBind (Zhu et al., 2024a)	Vicuna-7B v1.5 (Chiang et al., 2023)

A.2.2. MERV Encoder Details

Here, we detail the visual encoder details, LLM, and the training objectives. **We plan to release our full code including training and evaluation as well as all model checkpoints for the camera-ready version of the paper.**

LanguageBind We use the code from the original author, using the pre-trained weight `LanguageBind/LanguageBind_Video_merge` uploaded on `huggingface`.

Table 4. **Encoder Information**. Detailed information about the four encoders used in our experiments. They represent a broad coverage of visual information and training objectives.

Model	Architecture	Expertise	Training Datasets	Training Objective
LanguageBind (Zhu et al., 2024a)	ViT-L/14	Video+Language	VIDAL-10M, five-modal video examples	Contrastive
DINOv2 (Oquab et al., 2023)	ViT-L/14	Spatial	LVD-142M	Self-Supervised
ViViT (Arnab et al., 2021)	ViViT-B/16×2	Actions/Temporal	Kinetics-400/600, short videos	Supervised
SigLIP (Zhai et al., 2023)	ViT-B/16	Image+Language	4B curated image/text pairs	Contrastive

DINOv2 As DINOv2 is an image-model, we get embedding per frame, and concatenate them to be a video embedding. We use $\texttt{ViT}_{\texttt{Large}}$ model, pre-trained on LVD-142M dataset, and take the penultimate layer for the embeddings. Specifically, we use `timm`'s `vit_large_patch14_reg4_dinov2.lvd142m`

ViViT We use $\texttt{ViT}_{\texttt{base}}$ as our backbone, pre-trained on Kinetics-400 dataset. Specifically, we use `google/vivit-b-16x2-kinetics400` uploaded on `huggingface`. We use featurizer output as the video embedding.

SigLIP As SigLIP is an image-model, we get embedding per frame, and concatenate them to be a video embedding. We use $\texttt{ViT}_{\texttt{base}}$ as our backbone, and take the penultimate layer for the embeddings. Specifically, we use `timm`'s `vit_base_patch16_siglip_224`

We also considered multiple other options for encoders, such as CLIP-ViP (Xue et al., 2023) for our video-language contrastive expert, V-JEPA (Bardes et al., 2024) and Hiera (Ryali et al., 2023) for our pure video model, and CLIP (Radford et al., 2021) for our image-language contrastive expert, but found that our choices performed better overall.

A.3. Detailed Experimental Results

Here we tabulate the full experimental results that was abbreviated from the main paper. The first table (Table 5) ablates the different training recipes we tried for MERV, with extended discussion in Section 4.2.3.

Table 5. **Ablation of training stage recipes**. We explore different training recipe strategies, starting with the standard LLaVA recipe which Video-LLaVA adopted, along with some other variations.

Methods	MSVD-QA		MSRVTT-QA		TGIF-QA		Perception Test	ActivityNet-QA	
	Acc	Score	Acc	Score	Acc	Score	Acc	Acc	Score
MERV (frozen)	**70.97**	3.76	**59.03**	**3.25**	51.1	3.26	46.21	**50.87**	**3.34**
MERV, Video-LLaVA recipe	70.92	3.78	58.74	3.25	51.67	3.27	47.48	50.42	3.33
MERV (full)	70.48	**3.79**	57.25	3.24	**51.39**	**3.28**	**48.41**	49.93	3.33
MERV, mixed Stage 1+2	69.9	3.73	55.14	3.08	51.53	3.26	45.65	39.98	2.95

Pre-fusion projector details. The image-level projectors are similar to those described in MM-1 (McKinzie et al., 2024): 2D adaptive average pooling, a shallow attention resampler similar to a Perceiver Resampler (Alayrac et al., 2022), and convolutional pooling with 3 RegNet blocks on both sides of an average pool layer such as the C-Abstractor in Honeybee (Cha et al., 2024). For video-level projectors, we use a 3D average pool, where we pool to the same spatial dimension but furthermore pool the frame dimension by 2, and a 3D convolution where we add a single $2 \times 3 \times 3$ convolution before the same average pooling. For all projectors, we project to the same number of tokens $t \times h \times w$, using an adaptive average pool or $h \times w$ latent tokens for the attention resampler.

We also provide the full metric numbers in Table 6a, 6b and 6c for our ablations that are described in Sections 4.2.1 and 4.2.2.

Note on TGIF-QA Video-ChatGPT's and Video-LLaVA's author-reported numbers on TGIF are incomparable as they were on a subset of the dataset. See `https://github.com/PKU-YuanGroup/Video-LLaVA/issues/37`.

Table 6. **Full design choice ablation numbers.** Detailed experimental results of Tables 2a, 2b, 2c. We highlight our defaults in orange and **bold** the best results.

(a) **Pre-fusion projectors.** * is 16 frames instead of 8. Top two rows are projector-free baselines.

Projector	MSVD	MSRVTT	TGIF	Perc.	Params	FLOPs
257 tok	68.47	55.81	48.62	46.14	-	-
class tok	65.98	55	43.7	43.51	-	-
2D Avg	68.23	56.92	48.99	45.69	0	2.1M
2D Avg*	**69.08**	**58**	**50.01**	46.34	0	4.2M
2D Attn	65.76	55.23	43.35	44.14	12.7M	9.7G
2D Conv	67.48	56.78	47.6	45.04	237M	241G
3D Avg*	68.62	57.2	49.59	44.95	0	4.2M
3D Conv	68.56	57.03	49.28	**46.81**	113M	232G

(b) **Pre-fusion output token.** We ablate the optimal token size per frame for the pre-fusion projector.

Tkns	MSVD	MSRVTT	TGIF	Perc.
1	61.94	54.64	41.41	42.85
4	64.47	55.72	45.32	43.31
16	67.23	56.44	47.75	43.18
64	**69.08**	**58.00**	**50.01**	**46.34**
100	68.38	57.47	48.78	45.56
144	68.65	57.73	48.81	43.94
256	68.46	57.72	48.66	43.51

(c) **Feature fusion strategy.** We compare our feature fusion strategy with concatenating the visual embeddings in either token sequence dimension or the channel dimension, learning an optimal embedding mixture weights, and training with equal 25% mixture of visual embeddings.

Strategy	MSVD	MSRVTT	TGIF	Perc.	FLOPs
Cross-Attn	**70.97**	**59.03**	**51.1**	46.21	17.19 T
Concat (Seq.)	66.99	56.95	48.20	45.67	43.09 T
Concat (Ch.)	70.02	58.08	**51.1**	**47.36**	16.29 T
Learnable W	68.06	56.54	48.82	46.6	16.24 T
25% - Mixed	68.38	56.99	47.71	43.66	16.39 T

Table 7. **Effect of Each Encoder**. Detailed results of Figure 4a.

Methods	**MSVD-QA**		**MSRVTT-QA**		**TGIF-QA**		**Perception**	**ActivityNet-QA**		Avg
	Acc	Score	Acc	Score	Acc	Score	Acc	Acc	Score	Acc
All 4 encoders	**70.97**	**3.76**	**59.03**	**3.25**	**51.10**	**3.26**	46.21	50.87	**3.34**	**55.64**
w/o LanguageBind	68.52	3.69	57.10	3.19	50.20	3.23	45.23	49.78	3.31	54.17
w/o DINOv2	69.75	3.74	57.70	3.23	49.94	3.23	46.57	**51.43**	**3.34**	55.08
w/o ViViT	70.12	3.75	58.26	3.23	50.45	3.22	**46.94**	51.36	3.33	55.43
w/o SigLIP	69.85	3.74	57.55	3.22	50.27	3.22	46.20	50.06	3.32	54.79

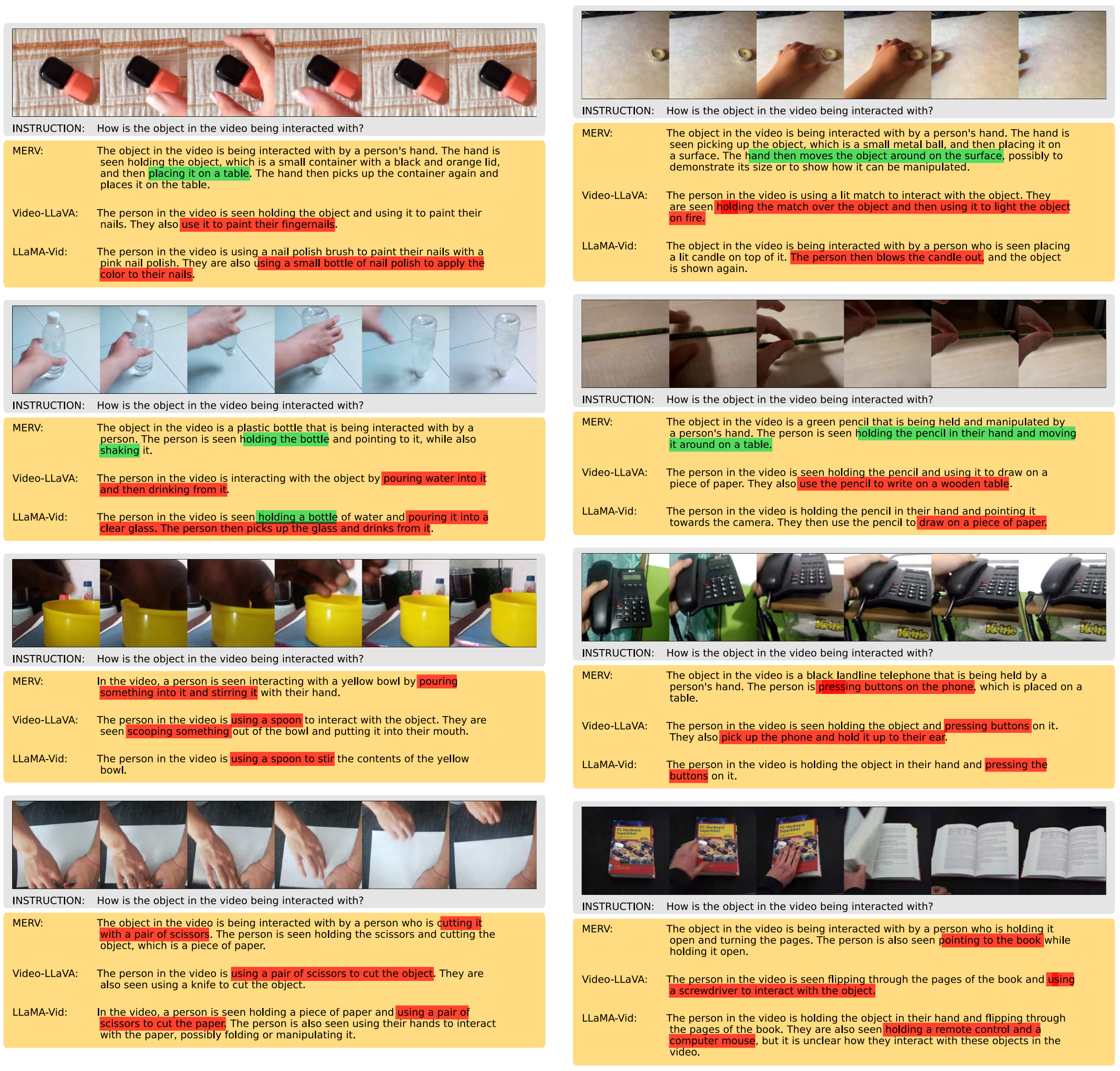
Table 8. **MERV Captures Single Encoder Performances**. Detailed experimental results of Figure 4a.

Methods	**MSVD-QA**		**MSRVTT-QA**		**TGIF-QA**		**Perception**	**ActivityNet-QA**		Avg		**Params**
	Acc	Score	Acc	Score	Acc	Score	Acc	Acc	Score	Acc	FLOPs	Overall
MERV	**70.97**	**3.76**	**59.03**	**3.25**	**51.10**	**3.26**	**46.21**	**50.87**	**3.34**	**55.64**	17.19 T	7686.0 M
LangBind	68.47	3.71	55.81	3.16	48.62	3.19	46.14	44.72	3.17	52.75	41.3 T	7147.0 M
DINOv2	65.44	3.62	53.46	3.09	41.53	2.96	42.73	43.39	3.09	49.31	40.88 T	7046.0 M
ViViT	59.95	3.43	51.81	3.05	38.1	2.84	40.2	43.98	3.16	46.81	27.12 T	6830.0 M
SigLIP	66.68	3.64	56.41	3.16	48.22	3.16	45.09	49.41	3.31	53.16	31.08 T	6834.0 M

Table 9. **Performance on WH-words**. Detailed experimental results of Figure 6.

	MSRVTT-what	MSRVTT-who	MSRVTT-how	MSRVTT-when	MSVD-what	MSVD-who	TGIF-what	TGIF-how	TGIF-where
MERV	**50.62**	**77.17**	83.96	72.23	**62.68**	**84.62**	**49.44**	**53.33**	**65.34**
ViViT	43.06	70.43	78.90	68.54	50.10	75.90	32.90	50.10	50.62
DINOv2	44.54	73.00	76.95	65.73	55.71	82.12	37.14	50.69	58.40
LanguageBind	46.89	74.86	83.41	**72.53**	59.66	82.95	46.23	53.24	59.92
SigLIP	47.96	74.41	**84.21**	71.20	57.17	82.23	45.65	53.25	60.21

A.4. Something-Something v2 Details


INSTRUCTION: How is the object in the video being interacted with?

MERV: The object in the video is being interacted with by a person's hand. The hand is seen holding the object, which is a small container with a black and orange lid, and then placing it on a table. The hand then picks up the container again and places it on the table.

Video-LLaVA: The person in the video is seen holding the object and using it to paint their nails. They also use it to paint their fingernails.

LLaMA-Vid: The person in the video is using a nail polish brush to paint their nails with a pink nail polish. They are also using a small bottle of nail polish to apply the color to their nails.

INSTRUCTION: How is the object in the video being interacted with?

MERV: The object in the video is a plastic bottle that is being interacted with by a person. The person is seen holding the bottle and pointing to it, while also shaking it.

Video-LLaVA: The person in the video is interacting with the object by pouring water into it and then drinking from it.

LLaMA-Vid: The person in the video is seen holding a bottle of water and pouring it into a clear glass. The person then picks up the glass and drinks from it.

INSTRUCTION: How is the object in the video being interacted with?

MERV: In the video, a person is seen interacting with a yellow bowl by pouring something into it and stirring it with their hand.

Video-LLaVA: The person in the video is using a spoon to interact with the object. They are seen scooping something out of the bowl and putting it into their mouth.

LLaMA-Vid: The person in the video is using a spoon to stir the contents of the yellow bowl.

INSTRUCTION: How is the object in the video being interacted with?

MERV: The object in the video is being interacted with by a person who is cutting it with a pair of scissors. The person is seen holding the scissors and cutting the object, which is a piece of paper.

Video-LLaVA: The person in the video is using a pair of scissors to cut the object. They are also seen using a knife to cut the object.

LLaMA-Vid: In the video, a person is seen holding a piece of paper and using a pair of scissors to cut the paper. The person is also seen using their hands to interact with the paper, possibly folding or manipulating it.

INSTRUCTION: How is the object in the video being interacted with?

MERV: The object in the video is being interacted with by a person's hand. The hand is seen picking up the object, which is a small metal ball, and then placing it on a surface. The hand then moves the object around on the surface, possibly to demonstrate its size or to show how it can be manipulated.

Video-LLaVA: The person in the video is using a lit match to interact with the object. They are seen holding the match over the object and then using it to light the object on fire.

LLaMA-Vid: The object in the video is being interacted with by a person who is seen placing a lit candle on top of it. The person then blows the candle out, and the object is shown again.

INSTRUCTION: How is the object in the video being interacted with?

MERV: The object in the video is a green pencil that is being held and manipulated by a person's hand. The person is seen holding the pencil in their hand and moving it around on a table.

Video-LLaVA: The person in the video is seen holding the pencil and using it to draw on a piece of paper. They also use the pencil to write on a wooden table.

LLaMA-Vid: The person in the video is holding the pencil in their hand and pointing it towards the camera. They then use the pencil to draw on a piece of paper.

INSTRUCTION: How is the object in the video being interacted with?

MERV: The object in the video is a black landline telephone that is being held by a person's hand. The person is pressing buttons on the phone, which is placed on a table.

Video-LLaVA: The person in the video is seen holding the object and pressing buttons on it. They also pick up the phone and hold it up to their ear.

LLaMA-Vid: The person in the video is holding the object in their hand and pressing the buttons on it.

INSTRUCTION: How is the object in the video being interacted with?

MERV: The object in the video is being interacted with by a person who is holding it open and turning the pages. The person is also seen pointing to the book while holding it open.

Video-LLaVA: The person in the video is seen flipping through the pages of the book and using a screwdriver to interact with the object.

LLaMA-Vid: The person in the video is holding the object in their hand and flipping through the pages of the book. They are also seen holding a remote control and a computer mouse, but it is unclear how they interact with these objects in the video.

Figure 8. **Samples of MERV in SSv2.** Due to our design, our method shows better temporal action understanding than other VideoLLMs. (Top two rows) However, due to the difficulty of the task, we see failure cases for VideoLLMs. (Bottom two rows)

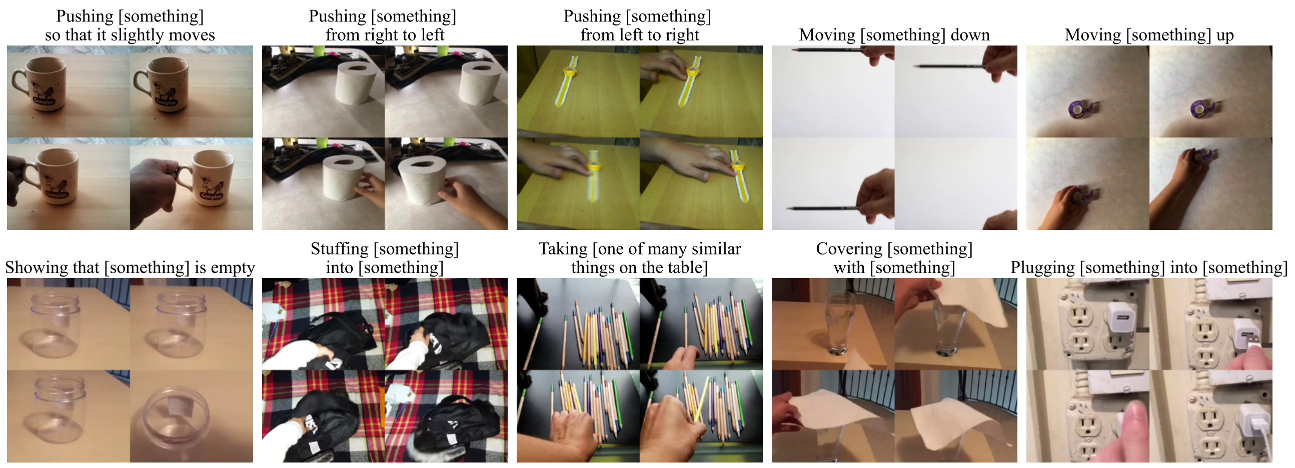

Pushing [something] so that it slightly moves
Pushing [something] from right to left
Pushing [something] from left to right
Moving [something] down
Moving [something] up
Showing that [something] is empty
Stuffing [something] into [something]
Taking [one of many similar things on the table]
Covering [something] with [something]
Plugging [something] into [something]

Figure 9. **Example video of Something-Something V2**. We see that ViViT show better performance in classes where temporal movement is critical for solving the task (Top row), while SigLIP performs better when the action can be inferred from the image without temporal information (Bottom row).

A.4.1. SOMETHING-SOMETHING V2 - OPENENDED

Table 10. **Performance on Something-Something V2 - OpenEnded**. These are the performance in shown Figure 4b

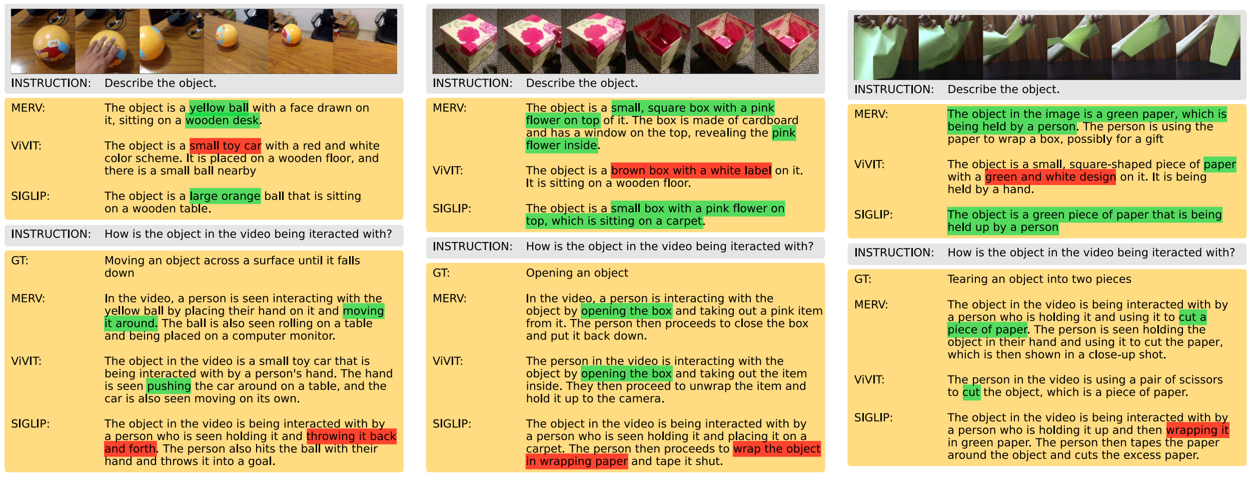
	MERV	MERV-Full	LanguageBind	DinoV2	ViViT	SigLIP	LLaMA-Vid-7B	LLaMA-Vid-13B	VideoLLaVA
Smth-Smth V2-OE-Temporal	6.82	**9.13**	3.63	3.88	5.50	4.25	6.07	3.94	5.57
Smth-Smth V2-OE	17.70	**20.65**	13.83	11.03	10.53	13.84	16.47	15.62	19.18
Smth-Smth V2-MCQ-Temporal	36.84	**40.65**	30.58	28.08	39.77	25.39	27.14	17.89	22.47
Smth-Smth V2-MCQ	**42.01**	39.76	36.82	33.06	26.78	34.86	36.63	39.43	23.14

Additionally, we evaluate Something-Something V2 as an open ended QA task, where the question is ”*How is the object in the video being interacted with?*”, and the answer is expected to be similar to the class label. We use Video-ChatGPT (Maaz et al., 2024)'s LLM evaluation for validating the VideoLLMs' output. Table 10 tabulates the results. We see similar conclusions as with MCQ where ViViT excels at temporal subset than other single-encoder LLMs, but failing to match the performance in full dataset. Nonetheless, our method shows the best performance overall.

A.4.2. SOMETHING-SOMETHING V2 - TEMPORAL

The 12 selected classes are as following:

- Approaching [something] with your camera
- Turning the camera downwards while filming [something]
- Turning the camera left while filming [something]
- Turning the camera right while filming [something]
- Turning the camera upwards while filming [something]
- Moving away from [something] with your camera
- Moving [something] away from the camera
- Moving [something] towards the camera
- Pulling [something] from left to right
- Pulling [something] from right to left
- Pushing [something] from left to right
- Pushing [something] from right to left



INSTRUCTION: Describe the object.

MERV: The object is a yellow ball with a face drawn on it, sitting on a wooden desk.

ViVIT: The object is a small toy car with a red and white color scheme. It is placed on a wooden floor, and there is a small ball nearby

SIGLIP: The object is a large orange ball that is sitting on a wooden table.

INSTRUCTION: How is the object in the video being iteracted with?

GT: Moving an object across a surface until it falls down

MERV: In the video, a person is seen interacting with the yellow ball by placing their hand on it and moving it around. The ball is also seen rolling on a table and being placed on a computer monitor.

ViVIT: The object in the video is a small toy car that is being interacted with by a person's hand. The hand is seen pushing the car around on a table, and the car is also seen moving on its own.

SIGLIP: The object in the video is being interacted with by a person who is seen holding it and throwing it back and forth. The person also hits the ball with their hand and throws it into a goal.

INSTRUCTION: Describe the object.

MERV: The object is a small, square box with a pink flower on top of it. The box is made of cardboard and has a window on the top, revealing the pink flower inside.

ViVIT: The object is a brown box with a white label on it. It is sitting on a wooden floor.

SIGLIP: The object is a small box with a pink flower on top, which is sitting on a carpet.

INSTRUCTION: How is the object in the video being iteracted with?

GT: Opening an object

MERV: In the video, a person is interacting with the object by opening the box and taking out a pink item from it. The person then proceeds to close the box and put it back down.

ViVIT: The person in the video is interacting with the object by opening the box and taking out the item inside. They then proceed to unwrap the item and hold it up to the camera.

SIGLIP: The object in the video is being interacted with by a person who is seen holding it and placing it on a carpet. The person then proceeds to wrap the object in wrapping paper and tape it shut.

INSTRUCTION: Describe the object.

MERV: The object in the image is a green paper, which is being held by a person. The person is using the paper to wrap a box, possibly for a gift

ViVIT: The object is a small, square-shaped piece of paper with a green and white design on it. It is being held by a hand.

SIGLIP: The object is a green piece of paper that is being held up by a person

INSTRUCTION: How is the object in the video being iteracted with?

GT: Tearing an object into two pieces

MERV: The object in the video is being interacted with by a person who is holding it and using it to cut a piece of paper. The person is seen holding the object in their hand and using it to cut the paper, which is then shown in a close-up shot.

ViVIT: The person in the video is using a pair of scissors to cut the object, which is a piece of paper.

SIGLIP: The object in the video is being interacted with by a person who is holding it up and then wrapping it in green paper. The person then tapes the paper around the object and cuts the excess paper.

Figure 10. **Example VideoLLM output on Something-Something v2**. While SigLIP performs better on object and scene recognition, it fails to understand temporal actions. ViViT fails on the details of object recognition, but has better understanding in temporal movements.

A.5. Additional Experiments and Analyses

A.5.1. Adding More Encoders

Additionally, we test with one more encoder Hiera (Ryali et al., 2023). The checkpoint we use was pretrained using the Masked Autoencoder (MAE) self-supervised learning technique on Kinetics 400 (K400), and then finetuned on K400 using supervised learning.

Table 11. **Performance with Hiera (Ryali et al., 2023).**

Methods	MSVD-QA		MSRVTT-QA		TGIF-QA		Perception	ActivityNet-QA	
	Acc	Score	Acc	Score	Acc	Score	Acc	Acc	Score
VideoLLaVA	67.74	3.69	56.90	3.18	47.99	3.17	47.08	3.27	44.22
MERV	**70.97**	**3.76**	**59.03**	**3.25**	**51.1**	**3.26**	50.87	**3.34**	46.21
MERV, ViViT replaced with Hiera	69.68	3.74	57.64	3.22	50.38	3.24	50.24	**3.34**	**47.50**
MERV + Hiera	69.67	3.72	58.26	3.23	50.32	3.22	**51.23**	**3.34**	46.23
ViViT Single Encoder LLM	59.95	3.43	51.81	3.05	38.10	2.84	43.98	3.16	40.20
Hiera Single Encoder LLM	55.38	3.28	49.21	2.95	36.02	2.76	44.01	3.15	40.20

The results are presented in the Table 11, with the best result in bold. Replacing ViViT with Hiera demonstrates improvements on the fine-grained spatio-temporal reasoning benchmark, Perception Test, with accuracy gain of 1.29%. Similarly, adding Hiera yields an improvement on ActivityNet, achieving a 0.36% increase in accuracy. However, on other benchmarks, the original MERV remains the strongest model. Overall, we observe no significant performance improvement when training with Hiera, which aligns with expectations, since Hiera is under the same paradigm as ViViT, functioning as a temporal expert trained on short videos. We also hypothesize that Hiera is more sensitive to the temporal stride than ViViT, as ViViT can reasonably deduce motion from uniformly sampled frames. We expect performance to improve if we incorporate encoders trained on different paradigms and data sources or process a much greater number of frames simultaneously, which we will leave for future work.

A.5.2. Attention Weights

Our analysis in Section 5.1 looks at how different cross-attention weights activate most on different videos, illustrated in Figure 11.

A.6. More Qualitative Results

We present additional qualitative results in Figure 12 and Figure 13.

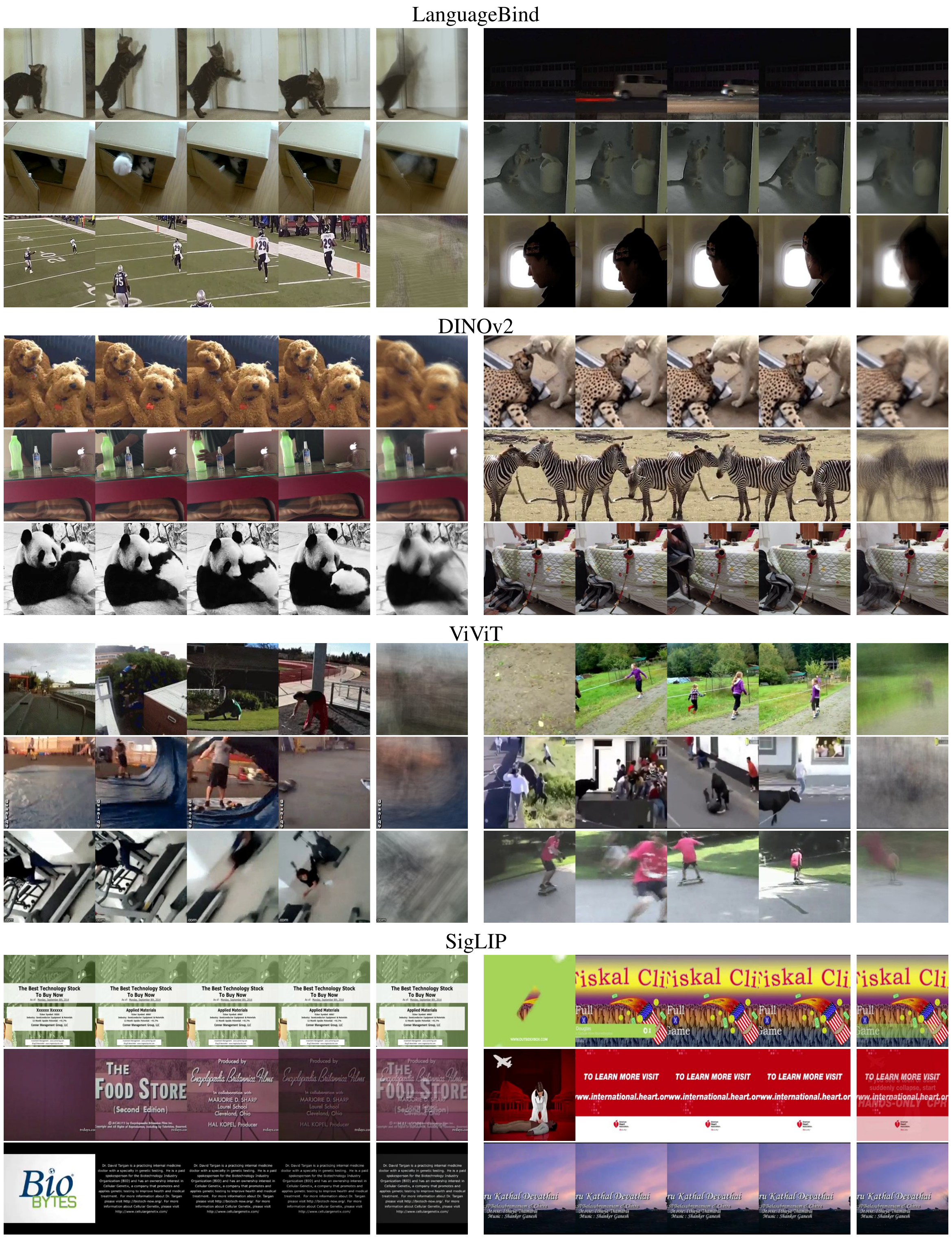
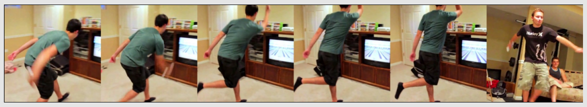


Figure 11. **Videos that give the highest attention weight for each of the encoders.** The right-most column shows the average frame of the video.



INSTRUCTION: Describe what the person is doing in the video.

MERV: The person in the video is playing a video game using a Wii remote. He is standing in a living room and swinging the remote around while playing.

INSTRUCTION: What game is he playing?

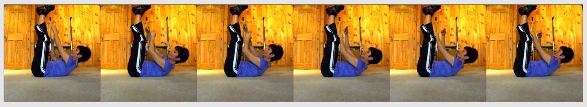
MERV: The man is playing a game of Wii bowling.

INSTRUCTION: What type of room is he in?

MERV: The man is in a living room.

INSTRUCTION: Is he playing alone?

MERV: No, he is not playing alone. There are other people in the room watching him play.



INSTRUCTION: Describe what the person is doing in the video.

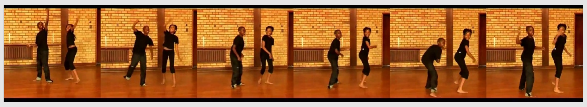
MERV: The person in the video is performing a leg stretch by lifting their right leg up and holding it in the air while lying on the floor.

INSTRUCTION: Describe the clothes that this person is wearing.

MERV: The person is wearing a blue shirt and black pants.

INSTRUCTION: What do you think this room is?

MERV: Based on the image, the room appears to be a gym or a workout area. A man is seen performing a workout routine, specifically a leg lift, on a carpeted floor. The presence of a wooden door and a bench in the room further supports the idea that this is a gym or a workout space. The man is wearing a blue shirt, which may indicate that he is part of a group or participating in a class. The room seems to be designed for physical activity and exercise, with the man using the bench and the carpeted floor to perform his workout routine.

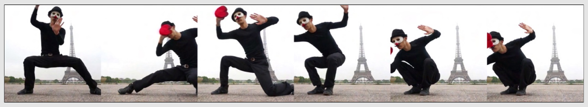


INSTRUCTION: Describe what the person is doing in the video.

MERV: The person in the video is dancing with a partner in a room. They are moving their arms and legs around while dancing.

INSTRUCTION: What is the texture of the wall?

MERV: The wall is made of bricks.



INSTRUCTION: Describe what the person is doing in the video.

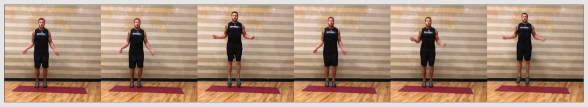
MERV: The person in the video is performing a dance routine while wearing a black hat and a red heart-shaped balloon. He is also wearing a black shirt and black pants.

INSTRUCTION: What do you see on the background?

MERV: In the background, there is a large Eiffel Tower.

INSTRUCTION: What country do you think this person is performing?

MERV: Based on the image, the person is performing in France, as they are wearing a black hat and kneeling in front of the Eiffel Tower. The Eiffel Tower is a famous landmark in Paris, France, and it is a popular tourist destination. The person's attire and the location suggest that they are performing in a French setting.

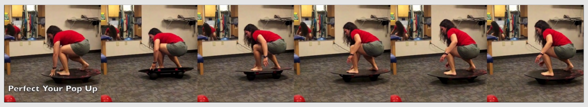


INSTRUCTION: Describe what the person is doing in the video.

MERV: The person in the video is demonstrating how to do a proper exercise on a mat. He is standing on the mat and talking to the camera while showing how to do the exercise.

INSTRUCTION: What is the color of the mat?

MERV: The color of the mat is red.



INSTRUCTION: Describe what the person is doing in the video.

MERV: The person in the video is performing a workout routine on a balance board. They are seen kneeling on the board and moving their arms and legs around while maintaining their balance.

INSTRUCTION: What is the color of the shirt that the person is wearing in the video?

MERV: The person in the video is wearing a red shirt.

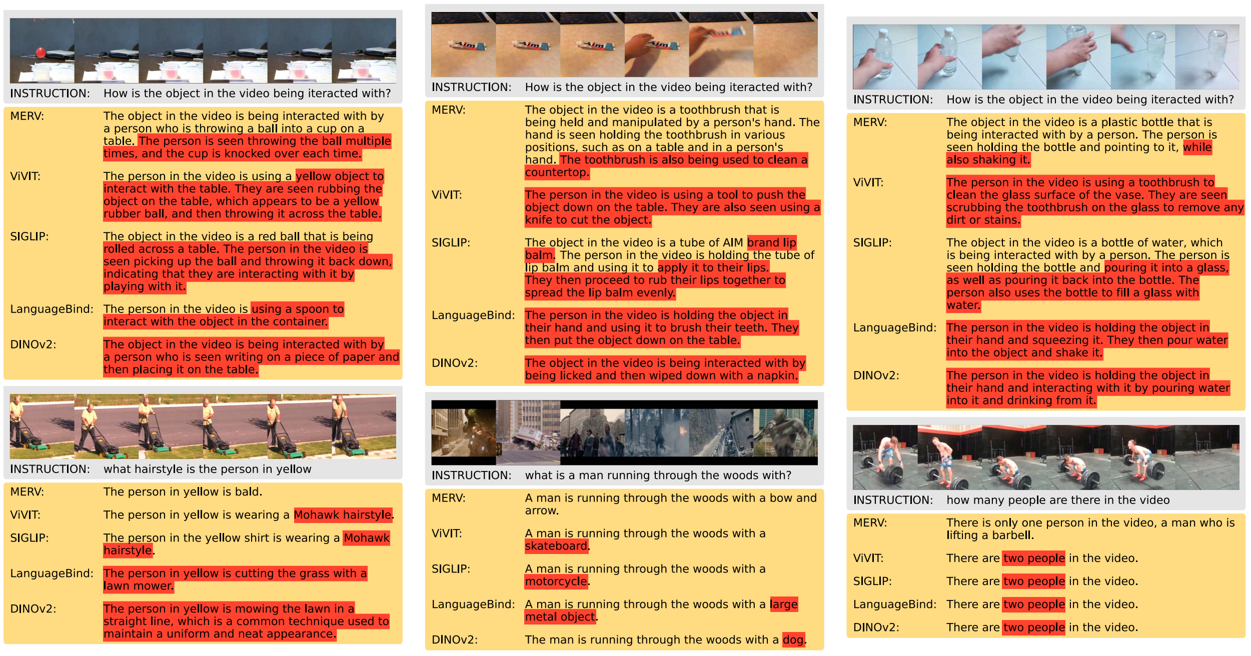
INSTRUCTION: What is the color of the balance board that the person is on?

MERV: The color of the balance board that the person is on is black.

INSTRUCTION: Describe the room that this person is inside of.

MERV: The person is inside a gym room, which is filled with various exercise equipment and gym mats.

Figure 12. Samples of MERV in video understanding.



INSTRUCTION:	How is the object in the video being iteracted with?
MERV:	The object in the video is being interacted with by a person who is throwing a ball into a cup on a table. The person is seen throwing the ball multiple times, and the cup is knocked over each time.
ViVIT:	The person in the video is using a yellow object to interact with the table. They are seen rubbing the object on the table, which appears to be a yellow rubber ball, and then throwing it across the table.
SIGLIP:	The object in the video is a red ball that is being rolled across a table. The person in the video is seen picking up the ball and throwing it back down, indicating that they are interacting with it by playing with it.
LanguageBind:	The person in the video is using a spoon to interact with the object in the container.
DINOv2:	The object in the video is being interacted with by a person who is seen writing on a piece of paper and then placing it on the table.

INSTRUCTION:	How is the object in the video being iteracted with?
MERV:	The object in the video is a toothbrush that is being held and manipulated by a person's hand. The hand is seen holding the toothbrush in various positions, such as on a table and in a person's hand. The toothbrush is also being used to clean a countertop.
ViVIT:	The person in the video is using a tool to push the object down on the table. They are also seen using a knife to cut the object.
SIGLIP:	The object in the video is a tube of AIM brand lip balm. The person in the video is holding the tube of lip balm and using it to apply it to their lips. They then proceed to rub their lips together to spread the lip balm evenly.
LanguageBind:	The person in the video is holding the object in their hand and using it to brush their teeth. They then put the object down on the table.
DINOv2:	The object in the video is being interacted with by being licked and then wiped down with a napkin.

INSTRUCTION:	How is the object in the video being iteracted with?
MERV:	The object in the video is a plastic bottle that is being interacted with by a person. The person is seen holding the bottle and pointing to it, while also shaking it.
ViVIT:	The person in the video is using a toothbrush to clean the glass surface of the vase. They are seen scrubbing the toothbrush on the glass to remove any dirt or stains.
SIGLIP:	The object in the video is a bottle of water, which is being interacted with by a person. The person is seen holding the bottle and pouring it into a glass, as well as pouring it back into the bottle. The person also uses the bottle to fill a glass with water.
LanguageBind:	The person in the video is holding the object in their hand and squeezing it. They then pour water into the object and shake it.
DINOv2:	The person in the video is holding the object in their hand and interacting with it by pouring water into it and drinking from it.

INSTRUCTION:	what hairstyle is the person in yellow
MERV:	The person in yellow is bald.
ViVIT:	The person in yellow is wearing a Mohawk hairstyle.
SIGLIP:	The person in the yellow shirt is wearing a Mohawk hairstyle.
LanguageBind:	The person in yellow is cutting the grass with a lawn mower.
DINOv2:	The person in yellow is mowing the lawn in a straight line, which is a common technique used to maintain a uniform and neat appearance.

INSTRUCTION:	what is a man running through the woods with?
MERV:	A man is running through the woods with a bow and arrow.
ViVIT:	A man is running through the woods with a skateboard.
SIGLIP:	A man is running through the woods with a motorcycle.
LanguageBind:	A man is running through the woods with a large metal object.
DINOv2:	The man is running through the woods with a dog.

INSTRUCTION:	how many people are there in the video
MERV:	There is only one person in the video, a man who is lifting a barbell.
ViVIT:	There are two people in the video.
SIGLIP:	There are two people in the video.
LanguageBind:	There are two people in the video.
DINOv2:	There are two people in the video.

Figure 13. **More qualitative results**. MERV tends to show improved understanding in temporal-heavy videos as in Something-Something v2 dataset (Goyal et al., 2017) (Top Row), while retaining the performance on scenic understanding, seen from popular video benchmarks (Xu et al., 2017; Yu et al., 2019) (Bottom Row).