

UNIFYING SPECIALIZED VISUAL ENCODERS FOR VIDEO LANGUAGE MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

The recent advent of Large Language Models (LLMs) has ushered sophisticated reasoning capabilities into the realm of video through Video Large Language Models (VideoLLMs). However, VideoLLMs currently rely on a single vision encoder for all of their visual processing, which limits the amount and type of visual information that can be conveyed to the LLM. Our method, *MERV*, Multi-Encoder Representation of Videos, instead leverages multiple frozen visual encoders to create a unified representation of a video, providing the VideoLLM with a comprehensive set of specialized visual knowledge. Spatio-temporally aligning the features from each encoder allows us to tackle a wider range of open-ended and multiple-choice video understanding questions and outperform prior state-of-the-art works. *MERV* is up to 3.7% better in accuracy than Video-LLaVA across the standard suite video understanding benchmarks, while also having a better Video-ChatGPT score. We also improve upon SeViLA, the previous best on zero-shot Perception Test accuracy, by 2.2%. *MERV* introduces minimal extra parameters and trains faster than equivalent single-encoder approaches by parallelizing the visual processing. Finally, we provide qualitative evidence that *MERV* successfully captures domain knowledge from each of its encoders. Our results offer promising directions in utilizing multiple vision encoders for comprehensive video understanding.

1 INTRODUCTION

Inspired by the sophisticated reasoning abilities of recent Large Language Models (LLMs) (8; 9; 41), researchers have focused on using them in many other domains to great success. The video counterparts, known as Video Large Language Models (VideoLLMs) (4; 28; 31; 37; 39; 62), connect pretrained 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.

Most multimodal LLMs, such as LLaVA (34) for images and Video-LLaVA (31) for videos, opt for contrastively pretrained encoders like CLIP (45) and LanguageBind (70). Their vision-language pretraining 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 (27). These encoders are almost always pretrained 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 (45) may be better suited with their multimodal semantic alignment, but are inferior to models such as DINOv2 (42) at fine-grained object level understanding. They also fail to take advantage of models trained specifically on videos, such as ViViT (2). 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 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 (2). Consequently, the reasoning capabilities of these VideoLLMs are directly limited by the inherent weaknesses of their respective pretrained encoders. Therefore, employing multiple encoders could allow us to complement one

		LB	DINOv2	ViViT	SigLIP	MERV		
054								
055								
056	Q: Is the order of the written letters the same as the order of the letters put on the table?		A. C. C. C. A.	✓	✗	✗	✗	✓
057	A: Yes B: I don't know C: No							
058								
059	Q: Is the camera moving or static?		A. B. A. A. B.	✗	✓	✗	✗	✓
060	A: moving B: static or shaking C: I don't know							
061								
062	Q: Was the first cup placed facing upwards or downwards?		A. A. B. A. B.	✗	✗	✓	✗	✓
063	A: upwards B: downwards C: I don't know							
064								
065	Q: Where is the person?		A. A. A. C. C.	✗	✗	✗	✓	✓
066	A: Kitchen B: Outdoor C: Living room or Bedroom							
067								

Figure 1: **Different visual experts exhibit individual strengths.** We show some examples where one single encoder model is the only model to correctly answer the Perception Test question (44).

encoder’s weaknesses with another encoder’s strengths. The wide adoption of the LLaVA paradigm is also indication that vision-language alignment is simple to achieve, even without language-aware vision models.

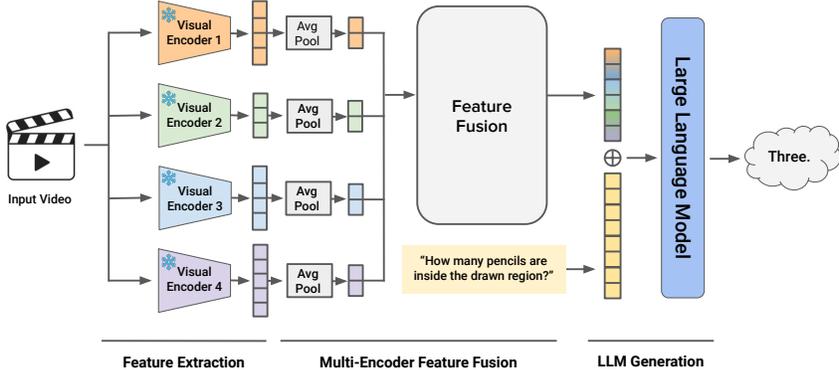
We propose *MERV*, a Multi-Encoder Representation of Videos, as a new method for integrating multiple visual encoders into a single VideoLLM using a cross-attentive encoder mixer for fusing representations. We introduce a spatio-temporally aligned representation for mixing the information from multiple 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 of the individual encoder methods, up to 3.7% better than prior works (31) on video reasoning benchmarks, i.e., from 47.1% to 50.8% on ActivityNet-QA (63), and on par with the state-of-the-art (62) on Perception Test (44), a challenging perception and reasoning diagnostic for video models. Finetuning the full model improves MERV past SeViLA (62) 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 (15). We show that MERV can accurately capture both the contrastive encoders’ (64; 70) strengths on general vision-language understanding, as well as ViViT’s (2) 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 (28) and Video-LLaMA (65) for chat-based video understanding, LLaMA-Adapter (66) for pre-alignment, Valley (37) with multilingual LLMs, InternVideo (56) with a dedicated video encoder training phase, and Video-ChatGPT (39) combining video-adapted encoders with LLMs. GPT4Video (57) supports video understanding and generation, while MovieChat (47) focuses on long video comprehension. Models like Chat-UniVi (20) and LLaMA-VID (30) optimize token usage for video representation. Other notable models include Vamos (55), which flexibly uses visual embeddings, action labels, and video captions as input; VideoChat2 (29), developed through three-stage progressive training; Video-LLaVA (31), which aligns image and video representations before projecting them to the LLM space; and VideoPrism (68), which also further trains a video encoder through masked distillation. Specialized models like VTimeLLM (18) focus on fine-grained video moment understanding and time-bound reasoning, while models like Elysium (54) and Merlin (61) can predict object trajectories. SeViLA (62) uses LLM for frame localizer of the video for multiple-choice tasks. Finally, recently LLaVA-Hound-DPO (67) explored using DPO and a higher quality training set for better instruction

¹Our code and pretrained weights will be made public for the camera-ready version of this paper.

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122



123
124
125
126
127

Figure 2: **Overview of MERV, a Multi-Encoder Representation of Videos.** 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 of the encoders, which is combined with the text query to produce our final generation.

128
129
130
131
132

following. Distinct from these aforementioned works, our approach centers on utilizing a diverse array of visual encoders, each with its own unique strengths and especially a video encoder, to significantly enhance the capabilities of the VideoLLM framework. By strategically utilizing these specialized encoders, we aim to capture a broader spectrum of visual information, thus enriching VideoLLMs’ understanding of video content.

133
134
135
136
137
138
139
140
141
142
143
144
145

Combining multiple encoders for multimodal LLMs is gaining attention. Eyes Wide Shut (51) explored mixing DINOv2 and CLIP features for LLaVA, but their results signal that mixing features effectively requires investigation. Both Mipha (71) and Prismatic-VLMs (22) 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 (14) and SPHINX (32) combines multiple image encoders by concatenating features along the channel dimension, while BRAVE (21) concatenates features from multiple encoders sequence-wise, followed by a QFormer with masked modeling. There is also the popular body of research on multimodal LLMs using many modalities including image, video, audio and/or 3D (7; 16; 17; 25; 33; 36; 38; 43; 48; 49; 65). In contrast, this paper dives into the video-language domain, exploring combining multiple image and video encoders and exploiting their structural similarities. Our feature fusion is both performant and efficient in FLOPs, and results in an all-encompassing additive mixture of features which previous works were unable to create without tradeoffs.

146 3 MERV: MULTI-ENCODER REPRESENTATION OF VIDEOS

147
148
149
150
151
152
153
154
155
156

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 (34; 35) paradigm (see Figure 2). Unlike previous works, our focus is not on combining multiple *modalities* (3; 70), but instead on combining multiple image and video encoders trained on different datasets and objectives. 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).

157 3.1 MULTI-ENCODER FEATURE EXTRACTION

158
159
160
161

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 Section 4.2. More details about these four encoders and other encoders we considered are in Appendix Table 4.

Spatial expert: DINOv2 (42) is trained using unsupervised learning on local-to-global correspondences in image data. The resulting features have a robust understanding of object parts, as well as semantic and low-level image understanding, but can suffer from poor language grounding.

Temporal expert: ViViT (2) 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 (64) 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 (70). Used by Video-LLaVA (31), LanguageBind is trained through joint multimodal learning between text and multiple modalities, including videos, infrared, and audio, 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 \mathcal{I} or videos \mathcal{V} , 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. Naively combining them is not possible as they all have different spatial shapes, and would also be prohibitively expensive at full resolution. We design a pre-fusion projector to both align and compress them.

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

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

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 $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1 \dots \bar{\mathbf{x}}_N] \in \mathbb{R}^{N \times d_L}$, where $\bar{\mathbf{x}}_e \in \mathbb{R}^d$ is each encoder’s features averaged over the sequence dimension ℓ for a faster computation, and N the number of encoders, and values as $\mathbf{X} = [\mathbf{x}_1 \dots \mathbf{x}_N] \in \mathbb{R}^{N \times \ell \times d}$. We calculate our final unified feature as

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

The final step is to concatenate the visual embedding and tokenized text together into the LLM. We use the base LLaMA-2 7B model (53), which we found performs better than the chat model. We test multiple alternate feature fusion strategies and their tradeoffs 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 the recent powerful advances in parallel processing for LLMs and use PyTorch’s Fully Sharded Data Parallel (69). As the video encoders themselves are much smaller than the LLM blocks and complete in around the same time, most of the overhead in running four encoders is already covered by having one encoder. 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 (22), 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 limited access testing. The Video-LLaVA codebase runs Stage 2 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 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)*, which performs only Stage 2 instruction tuning and achieves similar results to the original Video-LLaVA recipe in only 43% of the time, and *MERV (full)*, which unfreezes the LLM during Stage 1 as well 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

Datasets and training procedure. Our data mix is the same as Video-LLaVA (31). The Stage 1 data is single-turn concise captioning, with 558k (image, text) pairs from LAION filtered by LLaVA (34) and 702k (video, text) pairs from Valley (37). The Stage 2 data is multi-turn conversations, detailed captioning and reasoning, with 665k (image, text) pairs from LLaVA (34) and 100k (video, text) instructions from Video-ChatGPT (39).

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 (59), MSRVT-QA (59), TGIF (19), and ActivityNet-QA (63), as well as the multiple-choice benchmarks NExT-QA (58), VLEP (24), TVQA (23), and Perception Test (44). 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 (39) where

Methods	Visual Encoder And LLM	MSVD-QA		MSRVT-QA		TGIF-QA		Perception	ActivityNet-QA		NEXt-QA	VLEP	TVQA
		Acc	Score	Acc	Score	Acc	Score		Acc	Score			
<i>Alternative data mixes</i>													
Video-Chat (28)	(50), (11)	56.3	2.8	45.0	2.5	-	-	-	26.5	2.2	-	-	-
LLaMA-Adapter (66)	(45), (52)	54.9	3.1	43.8	2.7	-	-	-	34.2	2.7	-	-	-
Video-LLaMA (65)	(50; 26), (8)	51.6	2.5	29.6	1.8	-	-	-	12.4	1.1	-	-	-
Video-ChatGPT (39)	(45), (8)	64.9	3.3	49.3	2.8	-	-	-	35.2	2.7	-	-	-
SeViLA (62)	(50), (26)	-	-	-	-	-	-	46.2	-	-	63.6	64.4	38.2
LLaMA-VID-7B* (30)	(13), (8)	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* (30)	(13), (8)	70.25	3.77	58.58	3.26	51.26	3.26	41.54	46.79	3.23	60.03	61.98	41.33
<i>Same data mixes</i>													
Video-LLaVA* (31)	(70), (8)	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)	(70; 42; 2; 64), (53)	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	+0.07	+2.13	+0.07	+3.11	+0.09	+1.99	+3.79	+0.07	+3.48	-2.55	+4.62
MERV (full)	(70; 42; 2; 64), (53)	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	+0.10	+0.35	+0.06	+3.40	+0.11	+4.19	+2.85	+0.06	+1.75	-1.14	+1.76

Table 1: Comparison of different multimodal LLMs on video reasoning benchmarks. We employ ChatGPT to evaluate performance following Video-ChatGPT (39) where applicable (version gpt-3.5-turbo-0613). * denotes our evaluation of using the author provided checkpoint. The first five datasets were used as development sets; the last three were held-out for our final evaluation.

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 state-of-the-art works, including Video-LLaVA (31) that share our training data mixture, and other VideoLLMs (28; 30; 39; 62; 65; 66). 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 3.2% gain on MSVD and a 3.7% gain on ActivityNet. Both of our methods perform better overall than Video-LLaVA, even when using less data with just Stage 2 as shown by the MERV numbers. While MERV (full) is not a strict improvement to MERV, it still improves on some difficult benchmarks with its additional video-language alignment. We believe that outside of these testing sets, MERV (full) is a better model overall and recommend using this recipe when possible. Compared to LLaMA-VID (30), which uses a different training mix, we also better in nearly all benchmarks, up to around 4.5% across Perception Test, ActivityNet, and TVQA.² MERV (full) outperforms the previous state-of-the-art on the Perception Test zero-shot with 48.4%, compared to SeViLa (62) with a 46.2% accuracy. Overall, our design shows a significant improvement over Video-LLaVA and prior methods as a whole.

4.2 ABLATIONS

In this section, we justify the design choices for our architecture, covering our 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 pretrained 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. The image-level projectors are similar to those described in MM-1 (40): 2D adaptive average pooling, a shallow attention resampler similar to a Perceiver Resampler (1), and convolutional pooling with 3 RegNet blocks on both sides of an average pool layer like the C-Abstractor in Honeybee (6). 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

²Video-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>.

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

Projector	Avg Acc	Params	FLOPs	Tkns	MSVD	MSRVTT	TGIF	Strategy	Avg Acc	FLOPs
257 tok	54.76	-	-	1	61.94	54.64	41.41			
class tok	52.05	-	-	4	64.47	55.72	45.32			
2D Avg	54.96	0	2.1M	16	67.23	56.44	47.75	Cross-Attn	56.83	17.19 T
2D Avg*	55.86	0	4.2M	64	69.08	58.00	50.01	Concat (Seq.)	54.45	43.09 T
2D Attn	52.12	12.7M	9.7G	100	68.38	57.47	48.78	Concat (Ch.)	56.64	16.29 T
2D Conv	54.23	237M	241G	144	68.65	57.73	48.81	Learnable W	55.01	16.24 T
3D Avg*	55.09	0	4.2M	256	68.46	57.72	48.66	25% - Mixed	54.19	16.39 T
3D Conv	55.42	113M	232G							

(a) **Pre-fusion projectors.** * is 16 frames instead of 8. Top two rows are projector-free baselines. (b) **Pre-fusion output token.** We ablate the optimal token size per frame for the pre-fusion projector. (c) **Feature fusion strategy.** Cross-Attn additive mixing is best overall among all the strategies on accuracy, for its FLOPs. 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 results are in the Appendix.

pool or $h \times w$ latent tokens for the attention resampler. We report average performance across our development sets of MSVD, MSRVTT, TGIF, and Perception Test.

Pre-fusion projector. Table 2a tabulates each projector’s average accuracy, along with their parameter count and FLOPs, with LanguageBind as the single vision encoder and an 8 frame 64 token projection output by default. We find that 2D average pooling is the best overall projector, surpassing that of the full 257 token embedding (used in Video-LLaVA (31)) while also having no trainable parameters and the fewest FLOPs. The projection serves as a form of feature selection, allowing the LLM to efficiently reason only over the most relevant information. However, increasing the frame resolution from 8 to 16 was a large improvement, showing that increasing temporal resolution is still important. One result worth noting is the poor performance of the attentive resamplers, typically a popular projector choice. They are agnostic to structure, which leads them to being weaker projectors for us. 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 optimal output token size of the projector. Table 2b tabulates performance on different token output sizes when using 2D Average Pooling with 16 frames as the projector. We see that the performance peaks at 64 tokens, with worse performance for longer token lengths. This balances the number of tokens used for condensing the visual embedding while also minimizing the extra processing needed by the LLM, leading to the best overall performance.

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 projector for matching the LLM dimension. While sequence-wise concatenation is widely used in multimodal LLMs (51), our method outperforms it while using significantly less computation, with a 56.8% average accuracy compared to 54.4%, while also using $2.5\times$ fewer FLOPs. Concatenation channel-wise reaches a similar performance of 56.6% and a lightweight cost. However, our cross-attention shows slightly better performance, with the additional benefit of having accessible encoder weightings for analysis, so we do not choose channel-wise concatenation as 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 directly as a learnable scalar or by fixing the weights to be 0.25 for each of 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 given the visual input.

4.2.3 TRAINING RECIPES

Finally, we also compare different training recipes based on the literature and our own expertise. Traditional rule-of-thumb follows that of the original LLaVA recipe: a Stage 1 pre-training on captioning data to align the projectors only, and a Stage 2 instruction tuning on multi-turn complex reasoning data for both projectors and the LLM. Many recent works have attempted some combination of other strategies, such as unfreezing the vision encoders (12) or skipping the Stage 1 (22).

378 We systematically map out this landscape, fixing our dataset to be the same as Video-LLaVA’s and
 379 testing multiple hypotheses for the video domain. Contrary to Video-LLaVA, we found that the Stage
 380 1 phase did not have a significant effect on the final performance when training only the projectors
 381 and feature fusion, as can be seen in Table 5 in the Appendix. Performing only Stage 2 instruction
 382 tuning leads to similar results in 43% of the total time, so we adopt this recipe for efficiency. The
 383 general performances fluctuate, with the Perception accuracy of the Video-LLaVA recipe being
 384 slightly higher by 1.2%, but MSVD, MSRVT, and ActivityNet of ours are higher by around 0.4%.
 385 We refer to this recipe as *MERV (frozen)*.

386 This recipe is still unsatisfying as it leaves a large amount of data, approximately 1.3M vision-
 387 text pairs, unused for training. In our empirical observations, we often found that video-language
 388 alignment was not very strong. The distributions of language used in video datasets and benchmarks
 389 seem to sparsely overlap based on their sentence embeddings, which could have impacted our ability
 390 to perform well zero-shot on the downstream benchmarks. However, we found that if we *unfreeze* the
 391 LLM during Stage 1 and learn alignment between the LLM and the projectors and feature fusion, our
 392 performance improved on a few key benchmarks, especially Perception Test, by up to 2.2%.

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

401 Finally, we provide evidence for the efficiency
 402 of our method. We use the default FSDP shard-
 403 ing strategy PyTorch provides; it is not currently
 404 possible to specify explicit plans for which mod-
 405 ules go where (but may be possible as FSDP
 406 matures). However, even with this basic strat-
 407 egy, our method is dominated by the slowest
 408 single encoder present, incurring very little ad-
 409 ditional overhead from extra encoders due to this
 410 parallelization, making it cost efficient to scale up in the number of encoders.

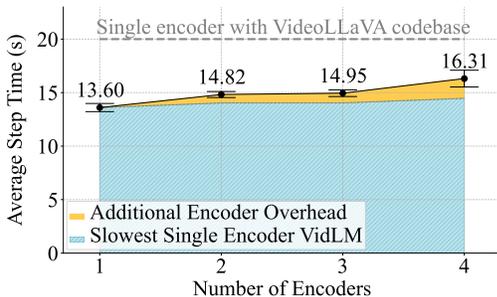


Figure 3: **Extra encoders incur minimal step time overhead.** Here we add encoders in the order of DINOv2, LanguageBind, SigLIP, ViViT, plotted alongside the slowest single encoder in each group.

5 ANALYSIS

5.1 STRENGTH IN THE ENSEMBLE OF ENCODERS

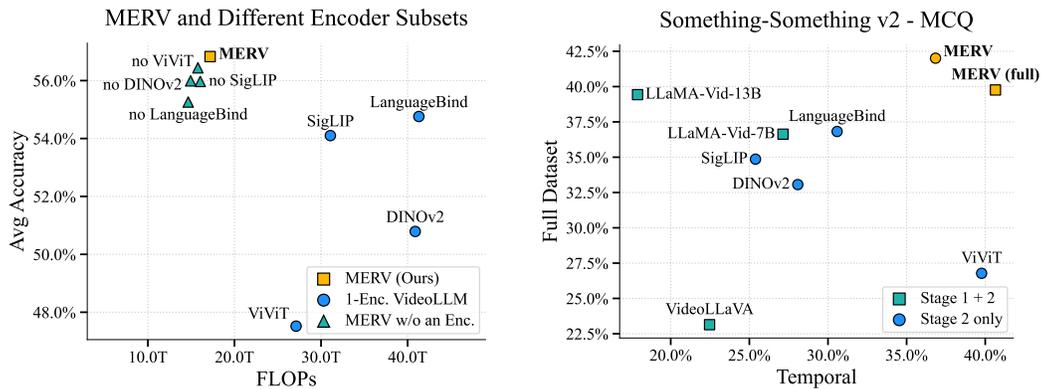
415 The original motivation of our work was to choose encoders with complementary visual knowledge
 416 to form a comprehensive representation for our final model. The key questions are 1) do we benefit
 417 by using more than one encoder, and 2) do we need all four encoders, i.e. does each one meaningfully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

419 **Can we make use of more encoders?** The conventional wisdom is to use a single encoder, typically
 420 a contrastively trained vision-language model like CLIP, SigLIP, or LanguageBind (45; 64; 70), in
 421 a VideoLLM. In Figure 4a, we show the four single encoder models corresponding to each of our
 422 chosen encoders using their full embeddings. They not only all perform worse than MERV but also
 423 use more FLOPs, as without our pre-fusion projectors, their sequence lengths are at least 4× ours.
 424

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

430 **Does MERV capture visual skills of different encoders?** Finally, we ask if our model effectively
 431 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

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485



(a) **Visual Encoder Subsets.** MERV outperforms single-encoder VideoLLMs, with our feature projectors unlocking more computational efficiency. Removing any encoder also reduces MERV performance. (b) **SSv2-MCQ and Temporal.** Temporal denotes performance on 12 selected classes where actions are indistinguishable if played in reverse.

Figure 4: **Analysis plots supporting our design of multiple encoders, from their accuracy to their skill specializations.** Average accuracy is across MSVD, MSRVTT, TGIF, and Perception Test. Full results are in the Appendix Tables 7, 8, 10.

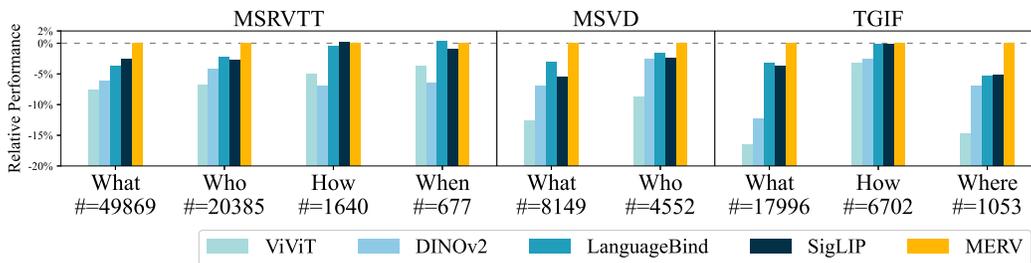


Figure 5: **Single encoder vs. MERV on different types of video tasks.** We plot the relative performance of VideoLLMs with different visual encoders. While each single encoder has its strength in different tasks, our method shows better performance than all the other single encoders in almost every task. We only plot tasks with more than 500 samples. See Appendix for details.

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 5 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 which combines different encoders into a unified representation that consistently matches or improves the best-performing encoder. Raw numbers are in Table 9 in the Appendix.

5.2 MERV CAN INTUIT MOTION AND GENERAL UNDERSTANDING SIMULTANEOUSLY

We take an alternate angle to quantifying how well our model learns from each of its individual encoders by looking towards classic video action recognition, from which we create detailed categories of skills based on the class names. We turn to Something-Something v2 (15) (SSv2) dataset where the goal of the original benchmark is to classify the 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. We provide qualitative examples where MERV is able to simultaneously provide descriptions and motion understanding, where it tends to align with either SigLIP (first row) or SigLIP (second row) based on the task (Figure 11). However, evaluating SSv2 as a zero-shot VideoLLM task is difficult with many specific categories. We repurpose the dataset as a 5-choice multiple-choice question (MCQ) dataset

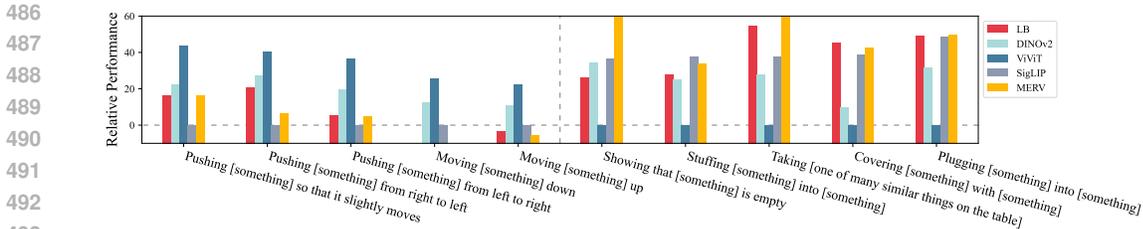


Figure 6: **Single-Encoder Performance Difference in Something-Something v2 - MCQ.** ViViT shows better performance on tasks where temporal understanding is crucial, while LanguageBind and SigLIP show better performance where task can be solved from single-frame understanding.

and fix the prompt to be "How is the object in the video being interacted with?". Incorrect choices were randomly sampled from the 173 other classes. We call this benchmark "Something-Something v2 - MCQ" to distinguish it from the original classification task.

We also 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*. 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 ViViT, which often falls short in other Video QA benchmarks, surprisingly performs better than other encoders at 39.77%, which is 9.19% higher than the next closest model LanguageBind. However for the full dataset, 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 on most other classes.

We plot the performances of 10 SSv2 classes where the performance difference between ViViT and SigLIP is largest in Figure 6. We see that actions that cannot be inferred from a single frame are the ones that ViViT performs better, e.g., *Moving [something] down* is indistinguishable from *Moving [something] up* if temporal information is omitted. Meanwhile, SigLIP performs better for classes where understanding the semantics of the scene can hint the action that is happening, e.g., if the video contains a cup and a bottle of water, one can easily expect *Showing [something] is empty* without watching the full video. See Appendix Figure 9 for sample videos of the 10 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 (31) and the 7B and 13B variants of LLaMA-Vid (30).

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 3.79% 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 are especially excited about this trend as it could allow our model to scale visual processing with the number of GPUs with better sharding strategies. We can place one expert on each device and obtain visual features in parallel while still retaining similar runtimes to having just one expert. We hope that this inspires others to also consider this problem as potentially another direction for scaling and improving their VideoLLMs.

REFERENCES

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 2022. 6
- [2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 1, 2, 4, 6, 17
- [3] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimaec: Multi-modal multi-task masked autoencoders. In *ECCV*, 2022. 3
- [4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 1
- [5] Adrien Bardes, Quentin Garrido, Jean Ponce, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. Revisiting feature prediction for learning visual representations from video. *arXiv:2404.08471*, 2024. 18
- [6] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *CVPR*, 2024. 6
- [7] Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages. *arXiv*, 2023. 3
- [8] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. 1, 6, 17
- [9] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 2023. 1
- [10] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024. 17
- [11] StableLM contributors. Stablelm: Stability ai language models. 6, 17
- [12] Haiwen Diao, Yufeng Cui, Xiaotong Li, Yueze Wang, Huchuan Lu, and Xinlong Wang. Unveiling encoder-free vision-language models. *arXiv*, 2024. 7
- [13] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*, 2023. 6, 17
- [14] Peng Gao, Renrui Zhang, Chris Liu, Longtian Qiu, Siyuan Huang, Weifeng Lin, Shitian Zhao, Shijie Geng, Ziyi Lin, Peng Jin, et al. Sphinx-x: Scaling data and parameters for a family of multi-modal large language models. *arXiv*, 2024. 3
- [15] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thureau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *arXiv*, 2017. 2, 9, 17
- [16] Jiaming Han, Kaixiong Gong, Yiyuan Zhang, Jiaqi Wang, Kaipeng Zhang, Dahua Lin, Yu Qiao, Peng Gao, and Xiangyu Yue. Onellm: One framework to align all modalities with language. In *CVPR*, 2024. 3
- [17] Jiaming Han, Renrui Zhang, Wenqi Shao, Peng Gao, Peng Xu, Han Xiao, Kaipeng Zhang, Chris Liu, Song Wen, Ziyu Guo, et al. Imagebind-llm: Multi-modality instruction tuning. *arXiv*, 2023. 3

- 594 [18] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to
595 grasp video moments. *arXiv*, 2023. 2
- 596 [19] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward
597 spatio-temporal reasoning in visual question answering. In *CVPR*, pages 2758–2766, 2017. 5
- 598 [20] Peng Jin, Ryuichi Takanobu, Caiwan Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified
599 visual representation empowers large language models with image and video understanding.
600 *arXiv*, 2023. 2
- 601 [21] Oğuzhan Fatih Kar, Alessio Tonioni, Petra Poklukar, Achin Kulshrestha, Amir Zamir, and
602 Federico Tombari. Brave: Broadening the visual encoding of vision-language models. *arXiv*,
603 2024. 3
- 604 [22] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa
605 Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models.
606 In *ICML*, 2024. 3, 5, 7
- 607 [23] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video
608 question answering. In *EMNLP*, 2018. 5
- 609 [24] Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. What is more likely to happen next?
610 video-and-language future event prediction. *arXiv*, 2020. 5
- 611 [25] Bo Li, Yuanhan Zhang, Liangyu Chen, Jinghao Wang, Jingkang Yang, and Ziwei Liu. Otter: A
612 multi-modal model with in-context instruction tuning. *arXiv*, 2023. 3
- 613 [26] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image
614 pre-training with frozen image encoders and large language models. *arXiv*, 2023. 6, 17
- 615 [27] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-
616 image pre-training for unified vision-language understanding and generation. In *International
617 Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 1
- 618 [28] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin
619 Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv*, 2023. 1, 2, 6, 17
- 620 [29] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo
621 Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark.
622 *arXiv*, 2023. 2
- 623 [30] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large
624 language models. *arXiv*, 2023. 2, 6, 10, 17
- 625 [31] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united
626 visual representation by alignment before projection. *arXiv*, 2023. 1, 2, 4, 5, 6, 7, 10, 17
- 627 [32] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin,
628 Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual
629 embeddings for multi-modal large language models. *arXiv*, 2023. 3
- 630 [33] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee.
631 Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. 3
- 632 [34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In
633 *NeurIPS*, 2023. 1, 3, 5
- 634 [35] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and
635 Noam Shazeer. Generating wikipedia by summarizing long sequences. *arXiv*, 2018. 3
- 636 [36] Shikun Liu, Linxi Fan, Edward Johns, Zhiding Yu, Chaowei Xiao, and Anima Anandkumar.
637 Prism: A vision-language model with multi-task experts. *Transactions on Machine Learning
638 Research*, 2024. 3

- 648 [37] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Minghui Qiu, Pengcheng Lu, Tao Wang,
649 and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability. *arXiv*,
650 2023. 1, 2, 5
- 651 [38] Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du,
652 Shuming Shi, and Zhaopeng Tu. Macaw-llm: Multi-modal language modeling with image,
653 audio, video, and text integration. *arXiv*, 2023. 3
- 654 [39] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt:
655 Towards detailed video understanding via large vision and language models. *arXiv*, 2023. 1, 2,
656 5, 6, 17, 19
- 657 [40] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp
658 Duffer, Dhruvi Shah, Xianzhi Du, Futang Peng, Floris Weers, Anton Belyi, Haotian Zhang,
659 Karanjeet Singh, Doug Kang, Ankur Jain, Hongyu Hè, Max Schwarzer, Tom Gunter, Xiang
660 Kong, Aonan Zhang, Jianyu Wang, Chong Wang, Nan Du, Tao Lei, Sam Wiseman, Guoli Yin,
661 Mark Lee, Zirui Wang, Ruoming Pang, Peter Grasch, Alexander Toshev, and Yinfei Yang. Mm1:
662 Methods, analysis & insights from multimodal llm pre-training, 2024. 6
- 663 [41] R OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5), 2023. 1
- 664 [42] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil
665 Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell
666 Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat,
667 Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal,
668 Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features
669 without supervision, 2023. 1, 4, 6, 17
- 670 [43] Artemis Panagopoulou, Le Xue, Ning Yu, Junnan Li, Dongxu Li, Shafiq Joty, Ran Xu, Silvio
671 Savarese, Caiming Xiong, and Juan Carlos Niebles. X-instructblip: A framework for aligning
672 x-modal instruction-aware representations to llms and emergent cross-modal reasoning. *arXiv*,
673 2023. 3
- 674 [44] Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Contente, Larisa Markeeva,
675 Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch,
676 Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael
677 Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew
678 Zisserman, and João Carreira. Perception test: A diagnostic benchmark for multimodal video
679 models. In *NeurIPS*, 2023. 2, 5
- 680 [45] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agar-
681 wal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya
682 Sutskever. Learning transferable visual models from natural language supervision. In *ICML*,
683 2021. 1, 6, 8, 17, 18
- 684 [46] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav
685 Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, Jitendra Malik, Yanghao
686 Li, and Christoph Feichtenhofer. Hiera: A hierarchical vision transformer without the bells-and-
687 whistles. In *ICML*, volume 202 of *ICML'23*, pages 29441–29454, Honolulu, Hawaii, USA, July
688 2023. JMLR.org. 18
- 689 [47] Enxin Song, Wenhao Chai, Guan hong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu,
690 Xun Guo, Tian Ye, Yan Lu, Jenq-Neng Hwang, et al. Moviechat: From dense token to sparse
691 memory for long video understanding. *arXiv*, 2023. 2
- 692 [48] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to
693 instruction-follow them all. *arXiv*, 2023. 3
- 694 [49] Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma,
695 and Chao Zhang. Fine-grained audio-visual joint representations for multimodal large language
696 models. *arXiv*, 2023. 3
- 700
701

- 702 [50] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training
703 techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 6, 17
704
- 705 [51] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes
706 wide shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, 2024. 3, 7
707
- 708 [52] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timo-
709 thée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez,
710 Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation
711 Language Models, February 2023. 6, 17
712
- 713 [53] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei,
714 Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open
715 foundation and fine-tuned chat models. *arXiv*, 2023. 5, 6, 16
716
- 717 [54] Han Wang, Yongjie Ye, Yanjie Wang, Yuxiang Nie, and Can Huang. Elysium: Exploring
718 object-level perception in videos via mllm, 2024. 2
719
- 720 [55] Shijie Wang, Qi Zhao, Minh Quan Do, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Vamos:
721 Versatile action models for video understanding. *arXiv*, 2023. 2
722
- 723 [56] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang,
724 Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative
725 and discriminative learning. *arXiv*, 2022. 2
726
- 727 [57] Zhanyu Wang, Longyue Wang, Zhen Zhao, Minghao Wu, Chenyang Lyu, Huayang Li, Deng
728 Cai, Luping Zhou, Shuming Shi, and Zhaopeng Tu. Gpt4video: A unified multimodal large
729 language model for instruction-followed understanding and safety-aware generation. *arXiv*,
730 2023. 2
731
- 732 [58] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-
733 answering to explaining temporal actions. In *CVPR*, 2021. 5
734
- 735 [59] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang.
736 Video question answering via gradually refined attention over appearance and motion. In *ACM*
737 *Multimedia*, 2017. 5, 17
738
- 739 [60] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo.
740 Clip-vip: Adapting pre-trained image-text model to video-language representation alignment.
741 *arXiv*, 2022. 18
742
- 743 [61] En Yu, Liang Zhao, Yana Wei, Jinrong Yang, Dongming Wu, Lingyu Kong, Haoran Wei, Tiancai
744 Wang, Zheng Ge, Xiangyu Zhang, et al. Merlin: Empowering multimodal llms with foresight
745 minds. *arXiv*, 2023. 2
746
- 747 [62] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language
748 model for video localization and question answering. *NeurIPS*, 2024. 1, 2, 6, 17
749
- 750 [63] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao.
751 Activitynet-qa: A dataset for understanding complex web videos via question answering.
752 In *AAAI*, 2019. 2, 5, 17
753
- 754 [64] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for
755 language image pre-training. In *ICCV*, 2023. 2, 4, 6, 8, 17
- 756 [65] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language
757 model for video understanding. *arXiv*, 2023. 2, 3, 6, 17
758
- 759 [66] Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li,
760 Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init
761 attention. *arXiv*, 2023. 2, 6, 17
762
- 763 [67] Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu,
764 Chunyuan Li, Alexander Hauptmann, Yonatan Bisk, and Yiming Yang. Direct preference
765 optimization of video large multimodal models from language model reward, 2024. 2

756 [68] Long Zhao, Nitesh B Gundavarapu, Liangzhe Yuan, Hao Zhou, Shen Yan, Jennifer J Sun, Luke
757 Friedman, Rui Qian, Tobias Weyand, Yue Zhao, et al. Videoprism: A foundational visual
758 encoder for video understanding. *arXiv*, 2024. 2
759

760 [69] Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright,
761 Hamid Shojanazeri, Myle Ott, Sam Shleifer, Alban Desmaison, Can Balioglu, Pritam Damania,
762 Bernard Nguyen, Geeta Chauhan, Yuchen Hao, Ajit Mathews, and Shen Li. Pytorch fsdp:
763 Experiences on scaling fully sharded data parallel, 2023. 5, 16

764 [70] Bin Zhu, Bin Lin, Munan Ning, Yang Yan, Jiayi Cui, HongFa Wang, Yatian Pang, Wenhao
765 Jiang, Junwu Zhang, Zongwei Li, et al. Languagebind: Extending video-language pretraining
766 to n-modality by language-based semantic alignment. *arXiv*, 2023. 1, 2, 3, 4, 6, 8, 17
767

768 [71] Minjie Zhu, Yichen Zhu, Xin Liu, Ning Liu, Zhiyuan Xu, Chaomin Shen, Yaxin Peng, Zhicai
769 Ou, Feifei Feng, and Jian Tang. A comprehensive overhaul of multimodal assistant with small
770 language models. *arXiv*, 2024. 3
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

A APPENDIX / SUPPLEMENTAL MATERIAL

A.1 LIMITATIONS AND BROADER IMPACT.

Our works are based on LLaMA-2 7B model (53), 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 running multiple encoders, which can be computationally intensive and can lead to out-of-memory errors in limited resource settings. While FSDP (69) 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. 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.

A.2 QUALITATIVE RESULTS

 <p>INSTRUCTION: Describe what the person is doing in the video.</p> <p>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.</p> <p>INSTRUCTION: What game is he playing?</p> <p>MERV: The man is playing a game of Wii bowling.</p> <p>INSTRUCTION: What type of room is he in?</p> <p>MERV: The man is in a living room.</p> <p>INSTRUCTION: Is he playing alone?</p> <p>MERV: No, he is not playing alone. There are other people in the room watching him play.</p>	 <p>INSTRUCTION: Describe what the person is doing in the video.</p> <p>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.</p> <p>INSTRUCTION: What do you see on the background?</p> <p>MERV: In the background, there is a large Eiffel Tower.</p> <p>INSTRUCTION: What country do you think this person is performing?</p> <p>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.</p>
 <p>INSTRUCTION: Describe what the person is doing in the video.</p> <p>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.</p> <p>INSTRUCTION: Describe the clothes that this person is wearing.</p> <p>MERV: The person is wearing a blue shirt and black pants.</p> <p>INSTRUCTION: What do you think this room is?</p> <p>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.</p>	 <p>INSTRUCTION: Describe what the person is doing in the video.</p> <p>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.</p> <p>INSTRUCTION: What is the color of the mat?</p> <p>MERV: The color of the mat is red.</p>
 <p>INSTRUCTION: Describe what the person is doing in the video.</p> <p>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.</p> <p>INSTRUCTION: What is the color of the shirt that the person is wearing in the video?</p> <p>MERV: The person in the video is wearing a red shirt.</p> <p>INSTRUCTION: What is the color of the balance board that the person is on?</p> <p>MERV: The color of the balance board that the person is on is black.</p> <p>INSTRUCTION: Describe the room that this person is inside of.</p> <p>MERV: The person is inside a gym room, which is filled with various exercise equipment and gym mats.</p>	 <p>INSTRUCTION: Describe what the person is doing in the video.</p> <p>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.</p> <p>INSTRUCTION: What is the color of the shirt that the person is wearing in the video?</p> <p>MERV: The person in the video is wearing a red shirt.</p> <p>INSTRUCTION: What is the color of the balance board that the person is on?</p> <p>MERV: The color of the balance board that the person is on is black.</p> <p>INSTRUCTION: Describe the room that this person is inside of.</p> <p>MERV: The person is inside a gym room, which is filled with various exercise equipment and gym mats.</p>

Figure 7: Samples of MERV in video understanding

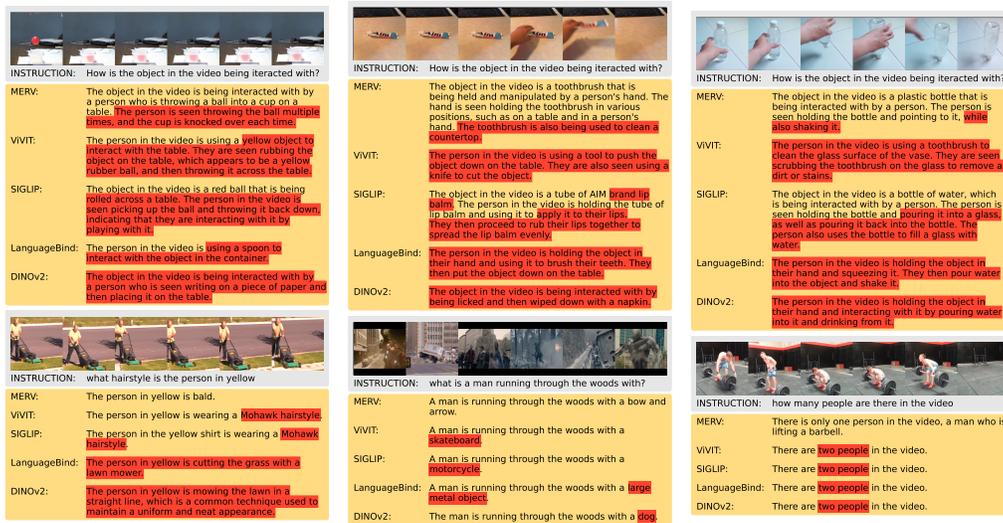


Figure 8: **MERV examples**. MERV tend to show improved understanding in temporal-heavy videos as in Something-Something v2 dataset (15) (Top Row), while retaining the performance on scenic understanding, seen from popular video benchmarks (59; 63) (Bottom Row).

B TRAINING DETAIL

B.0.1 BASELINE ENCODER AND LLM DETAILS

Model	Visual Encoder	LLM
Video-Chat (28)	ViT-G (EVA-CLIP) (50)	StableVicuna (11)
LLaMA-Adapter (66)	CLIP (45)	LLaMA-1 7B (52)
Video-LLaMA (65)	ViT-G (EVA-CLIP) (50) + BLIP-2 Q-Former (26)	Vicuna-7B v0 (8)
Video-ChatGPT (39)	CLIP (45)	Vicuna-7B v1.1 (8)
SeViLA (62)	ViT-G (EVA-CLIP) (50) + BLIP-2 Q-Former (26)	FlanT5-XL (3B) (10)
LLaMA-VID-7B (30)	EVA-G (13)	Vicuna-7B v1.5 (8)
LLaMA-VID-13B (30)	EVA-G (13)	Vicuna-13B v1.5 (8)
Video-LLaVA* (31)	LanguageBind (70)	Vicuna-7B v1.5 (8)

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

B.0.2 MERV ENCODER DETAILS

Model	Architecture	Expertise	Training Datasets	Training Objective
LanguageBind (70)	ViT-L/14	Video+Language	VIDAL-10M, five-modal video examples	Contrastive
DINOv2 (42)	ViT-L/14	Spatial	LVD-142M	Self-Supervised
ViViT (2)	ViViT-B/16×2	Actions/Temporal	Kinetics-400/600, short videos	Supervised
SigLIP (64)	ViT-B/16	Image+Language	4B curated image/text pairs	Contrastive

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.

Here, we detail the visual encoder details, LLM, and the training objectives. We plan to release the code 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.

DINOv2 As DINOv2 is an image-model, we get embedding per frame, and concatenate them to be a video embedding. We use ViT_{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_dinov2lvd142m

ViViT We use ViT_{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 ViT_{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 (60) for our video-language contrastive expert, V-JEPA (5) and HierA (46) for our pure video model, and CLIP (45) for our image-language contrastive expert, but found that our choices performed better overall.

B.1 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.

Methods	MSVD-QA		MSRVTT-QA		TGIF-QA		Perception	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

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.

We also provide the full numbers for our ablations in Sections 4.2.1 and 4.2.2.

Projector	MSVD	MSRVTT	TGIF	Perc.	Params	FLOPs	Tkns	MSVD	MSRVTT	TGIF	Perc.
257 tok	68.47	55.81	48.62	46.14	-	-	1	61.94	54.64	41.41	42.85
class tok	65.98	55	43.7	43.51	-	-	4	64.47	55.72	45.32	43.31
2D Avg	68.23	56.92	48.99	45.69	0	2.1M	16	67.23	56.44	47.75	43.18
2D Avg*	69.08	58	50.01	46.34	0	4.2M	64	69.08	58.00	50.01	46.34
2D Attn	65.76	55.23	43.35	44.14	12.7M	9.7G	100	68.38	57.47	48.78	45.56
2D Conv	67.48	56.78	47.6	45.04	237M	241G	144	68.65	57.73	48.81	43.94
3D Avg*	68.62	57.2	49.59	44.95	0	4.2M	256	68.46	57.72	48.66	43.51
3D Conv	68.56	57.03	49.28	46.81	113M	232G					

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

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

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

(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.

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.

B.2 SOMETHING SOMETHING V2 DETAILS

B.2.1 SOMETHING-SOMETHING V2 - OPENENDED.

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

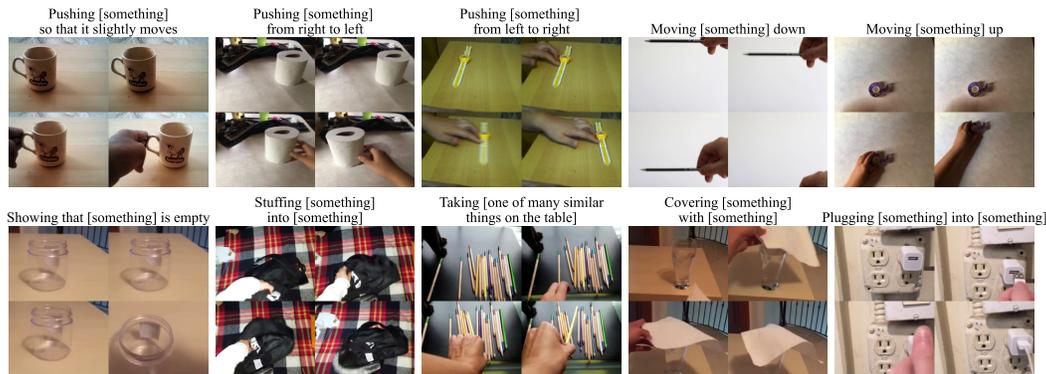
Methods	MSVD-QA		MSRVTT-QA		TGIF-QA		Perception	ActivityNet-QA		Avg
	Acc	Score	Acc	Score	Acc	Score	Acc	Acc	Score	Acc
All 4 encoders	70.97	3.76	59.03	3.25	51.10	3.26	46.21	50.87	3.34	55.64
w/o LanguageBind	68.52	3.69	57.10	3.19	50.20	3.23	45.23	49.78	3.31	54.17
w/o DINOv2	69.75	3.74	57.70	3.23	49.94	3.23	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 7: **Effect of Each Encoder**. Detailed results of Figure 4a

Methods	MSVD-QA		MSRVTT-QA		TGIF-QA		Perception		ActivityNet-QA		Avg	Params Overall
	Acc	Score	Acc	Score	Acc	Score	Acc	Score	Acc	Score	Acc	
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 8: **MERV Captures Single Encoder Performances**. Detailed experimental results of Figure 4a.

	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

Table 9: **Performance on WH-words**. Detailed experimental results of Figure 5Figure 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).

	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

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

the class label. We use Video-ChatGPT (39)’s LLM evaluation for validating the VideoLLMs’ output. Table 10 tabulates the results.

B.2.2 SOMETHING-SOMETHING V2 - TEMPORAL.

The 12 selected classes are as following:

1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079



Figure 10: 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)

- 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

1080			
1081			
1082	INSTRUCTION: Describe the object.	INSTRUCTION: Describe the object.	INSTRUCTION: Describe the object.
1083	MERV: The object is a yellow ball with a face drawn on it, sitting on a wooden floor .	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 .	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.
1084	VViT: 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.	VViT: The object is a brown box with a white label on it. It is sitting on a wooden floor.	VViT: 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.
1085	SIGLIP: The object is a large orange ball that is sitting on a wooden table.	SIGLIP: The object is a small box with a pink flower on top, which is sitting on a carpet .	SIGLIP: The object is a green piece of paper that is being held up by a person .
1086	INSTRUCTION: How is the object in the video being interacted with?	INSTRUCTION: How is the object in the video being interacted with?	INSTRUCTION: How is the object in the video being interacted with?
1087	GT: Moving an object across a surface until it falls down.	GT: Opening an object.	GT: Tearing an object into two pieces.
1088	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.	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.	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.
1089	VViT: 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.	VViT: 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.	VViT: The person in the video is using a pair of scissors to cut the object, which is a piece of paper.
1090	SIGLIP: The object in the video is being interacted with by a person who is seen holding it and blowing it back and forth . The person also hits the ball with their hand and throws it into a goal.	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.	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 11: **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.

C ADDITIONAL ANALYSIS

C.1 ATTENTION WEIGHTS

We look at the attention weight on our 4 benchmark datasets (MSRVTT, TGIF, MSVD, and Perception Test), and visualize the videos that have the highest attention weight for each of the encoders on Figure 12. As expected, ViViT attention weights are highest on videos with large motion, as ViViT have strong temporal motion understanding. Meanwhile, SigLIP, as they are vision-language contrastively trained, is preferred by videos that have textual data in the video. DINOv2 and Language-Bind are both preferred by videos with static scenes, but Language-Bind, as it is contrastively trained with video and language, is preferred by video with some foreground motion.

1134
 1135
 1136
 1137
 1138
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187

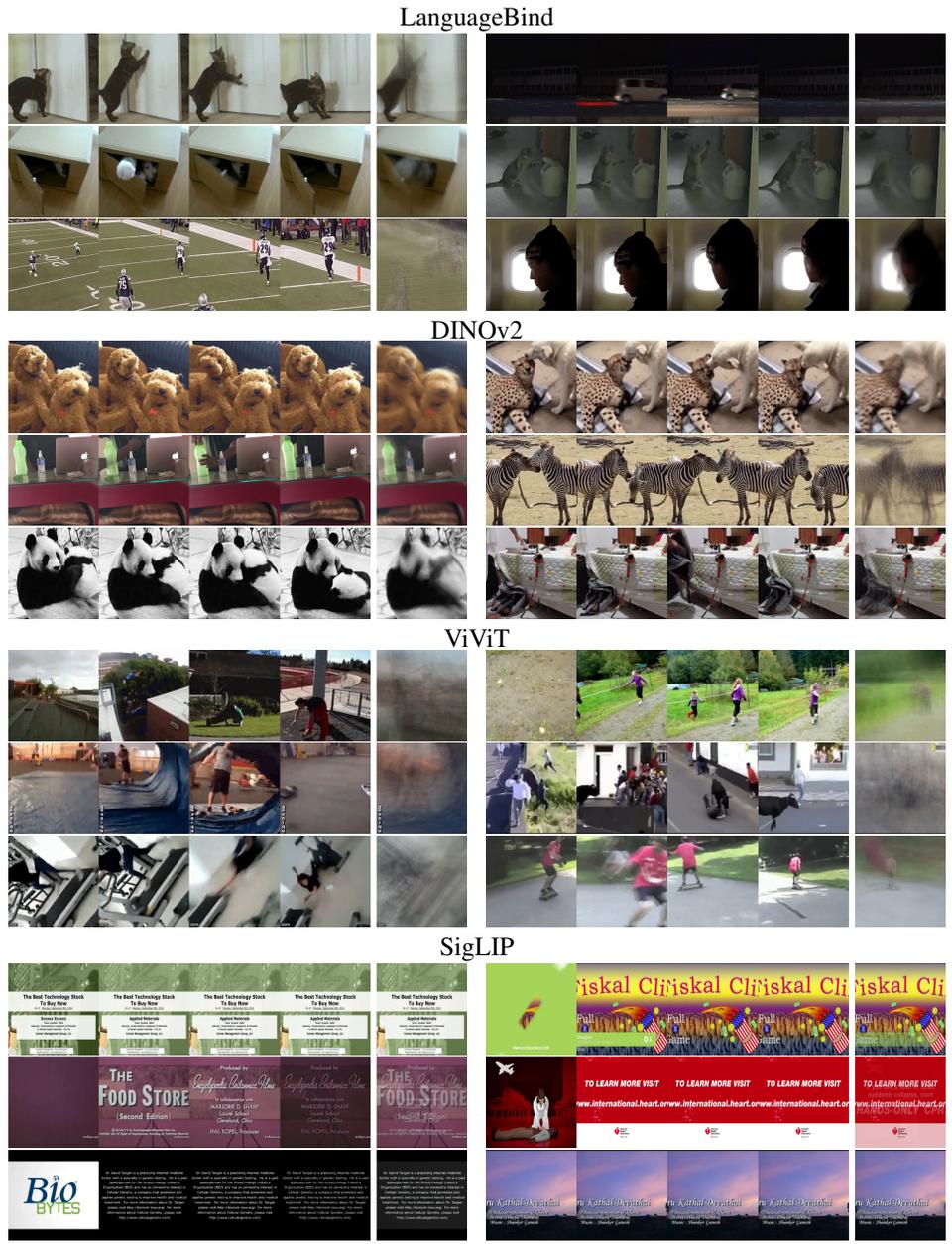


Figure 12: Videos that give the highest attention weight for each of the encoders. The right-most column shows the average frame of the video.