

Appendix

A Simple Recipe for Contrastively Pre-training Video-First Encoders Beyond 16 Frames

Acknowledgements

We thank Chuhan Zhang for her thoughtful feedback on early drafts of the work; Chris Dyer, Joao Carreira, and Gabriel Brostow for their support, feedback, and guidance through the project; and Lucia Lopez for her valuable help in generating synthetic video captions.

A. Methodology Details

Image-to-video architectural adaptations. ViViT with joint space-time attention does not add any new parameters to vanilla ViT, which facilitates transfer between image and video models. However, we change the (position and patch) embedding layers to adapt to video inputs. In particular, we extend position embeddings to longer context via repetition, hence tokens within a frame have the pre-trained position encodings and tokens across time have identical position information. We experimented with both repetition and interpolation for initialization and find that both approaches provide similar results when fine-tuned on video data, whereas repetition works better in a zero-shot setting. As mentioned in Section 3.1 we use 3D convolution for embedding tubelets from the input video. We initialize the convolution weights via the 2D image-based weights by “inflating” them, *i.e.* replicating the filters along the temporal dimension and performing mean pooling [2, 4, 7].

Multi-resolution Patchification and TubeViT. Piergiovanni et al. [20] tune multiple model variants of different sizes on different data mixtures (images vs. videos) in order to achieve good initialization of the multiresolution patch/tubelet embeddings. We aim to avoid additional pre-training steps with multiple pre-trained models and find that initializing all embedding layers with the image-based 2D weights works well in practice. Additionally, Piergiovanni et al. [20] handcraft fixed spatiotemporal position encodings to account for overlapping tubelets with different spatial and/or temporal strides, while showing that learnable position encodings lead to inferior performance. We overcome these obstacles and create a more generic approach that does not need handcrafting by employing factorized

attention for processing the different “views” of the video (*i.e.* overlapping parts of the video sampled at different spatiotemporal resolution). We find that this approach leads to better performance in contrast to flatten all multiresolution tubelets and feed them into the joint space-time attention as one long sequence. Following [20], we use four convolution layers with the specified kernels, strides, and offsets. In our exploration, we also try spatiotemporal kernels of (T, H, W) sizes: (4, 16, 16), (2, 32, 32), and (4, 32, 32).

Adapters and LoRA. We explored use of MLP Adapters [10] and LoRA [11]. For MLP Adapters, we add a bottleneck layer at every layer of the encoder (after the feed-forward block):

$$h_{down,i} = f(LN(W_d h_i + b_d)) \quad (1)$$

$$h_{up,i} = W_u h_{down,i} + b_u \quad (2)$$

$$h_i = h_i + h_{up,i}, \quad (3)$$

where $W_d \in \mathbb{R}^{d_m \times d_b}$, d_m is the model dimension, d_b is the bottleneck dimension (*i.e.* $\ll d_m$), $f(\cdot)$ is a non-linearity (we use *ReLU*) LN is a trainable layer normalization, $W_u \in \mathbb{R}^{d_b \times d_m}$, and h_i is the output of the feed-forward layer.

For LoRA, we decompose the linear QKV input projection, the output self-attention projection, and the dense feed-forward block of each layer in ViViT:

$$h = W_o x + \frac{1}{\alpha} B A x, \quad (4)$$

where W_o is the original weight matrix of each block that remains frozen while tuning the learnable B and A matrices, $B \in \mathbb{R}^{d_m \times r}$, $A \in \mathbb{R}^{r \times d_m}$, $r \ll d_m$ is the rank of the decomposition matrices, and α is a hyperparameter for easier tuning of the model, as recommended by [11].

For ViViT-B experiments: (1) for MLP Adapters, our adapter bottleneck dimension d_b was set to 384 and we zero-initialize weights W_d, W_u and biases b_d, b_u . (2) For LoRA, we use a $r=64$, and $\alpha = 1/64$ and same parameter initialization as [11]. For ViViT-L experiments: the positioning

and initialization of adapters were the same as with ViViT-B, but with d_b of 768 (for MLP adapters) and r of 128 (for LoRA).

Temporal pooling + Perceiver resampler. For video-to-text tuning, we use a Perceiver resampler with 3 layers, 1024 model hidden dimension, 8 heads for the multi-head attention blocks and 4096 inner-layer dimension for the feed-forward blocks. The Perceiver resampler was originally introduced by Alayrac et al. [1] in order to produce a fixed number of visual tokens to be fed into the LM independently of the output length of the visual encoder (e.g., for multiple images or videos). However, we empirically find that when we scale to videos beyond 16-32 frames, the Perceiver resampler becomes unstable during training leading to uniform attention distribution over visual tokens, even with appropriate Q/K cross-attention normalization and other tricks. In order to avoid unstable training for long frame sequences, we first average pool visual tokens across the temporal dimension in order to have a fixed number of tokens independently of the video length and then apply Perceiver resampler utilizing the same number of latent queries as the number of input tokens (i.e., 256 in our case for frames of 256x256 spatial resolution and convolution kernel of 16x16 spatial dimensions). Using this combination of a Perceiver resampler and temporal pooling leads to the best performance, compared to alternatives, although performance gains are generally small in comparison to completely removing the Perceiver resampler. We leave to future work the exploration of better ways to project and feed visual tokens into frozen LMs that can scale well with sequence length.

B. Implementation Details

In this section, we discuss implementation details including training specifications, compute resources, synthetic data for training on short and long videos, and evaluation details per benchmark.

Training details. We consider input frames with a 256x256 spatial resolution and patchify videos with a default convolution kernel of 2x16x16. We center-crop all frames and do not consider additional data augmentation or regularization methods. We present our experimental settings categorized by stage of training on Table 1. We report compute requirements for training model variants on Table 2. For offline evaluation, on all tasks and datasets, we used four TPUv5 chips.

Synthetic training data. As mentioned in Section 5, we find that VideoCC3M presents poor video-text align-

ment and compromises performance of our models. For this reason, we experiment with generating synthetic textual descriptions of the videos via PALI-3 [6]. Videos in VideoCC3M are at 10FPS and on average only 10 seconds long and static; differences between consecutive frames are small. We roughly sample the center frame from each video and feed it into PALI for generating a detailed description (i.e. selecting the 50th frame, and if this fails, selecting the 25th frame, or if this fails, the 0th frame). We empirically find that (1) generated captions are more accurate than the original, automatically mapped ones, and (2) PALI-3 is able to generate long and detailed captions that mention several details present in the video. We show the effect of adding the PALI-captioned version of VideoCC3M in Appendix C.

Moreover, we use the full-length videos of HowTo100M for training LONGViViT on longer contexts (HowTo100M Summary; Section 5). The full-length videos have an average duration of 6.5 minutes and are accompanied by ASR, i.e. automatic closed captions of people describing their actions. However, instead of directly using ASR, which can be noisy, not coherent between utterances, and contains irrelevant information and comments from the speakers, we use a LLM, namely Chinchilla [9], for better cleaning and summarizing the ASR. We further filter generated summaries to discard repetitions. The resulting summaries are more coherent, condensed and describe the desired task and accompanied actions.

Evaluation details per benchmark. For our ablation studies on text-video retrieval (Section 6.1) we use the validation sets of all benchmarks. We follow the settings of Yan et al. [26], when applicable, for reporting our main results in Section 6.2. Specifically:

- *MSR-VTT*: We report results in Table 1 of the main paper on the full test set for text-video retrieval and on a subset of 1000 examples from the test set for video captioning. Each video is accompanied by 20 different captions, so we report average over the different captions.
- *ActivityNet Captions*: We report results on the `val1` subset. For short video evaluation in Table 2 of the main paper, we consider the first 16 seconds (sampling frames at 1 FPS from the beginning of the video) of the 180-second videos for paragraph-video retrieval and segment-by-segment captions for video captioning. For long video evaluation in Table 3 of the main paper, we consider the raw, full-length videos of `val1` without ground-truth segmentation. There are videos with multiple captions, in which case we report the average.
- *YouCook2*: We report results on the ground-truth segments and full-length videos of the validation set in Tables 2 and 3 of the main paper, respectively.
- *VATEX*: We report results on the validation set for text-video retrieval and on the test set for video captioning.

	Image-to-Video Contrastive Pre-training	Short-to-long Video Contrastive Pre-training	Video-to-text Tuning	Dataset-specific Fine-tuning
Optimizer		AdamW		
Learning rate schedule		Cosine with linear warmup		
Gradient clip		2.0		1.0
Weight decay rate		1e-2		1e-4
Batch size		512	128	64-128
Base learning rate		5e-5	4e-5	1-5e-6
Linear warmup steps		1k	2k	1k
Training steps	800k	50k	80k	10k
Training steps	800k	50k	80k	10k

Table 1. Training specifications for (1) image-to-video contrastive pre-training, (2) short-to-long video contrastive tuning, (3) video-to-text tuning, and (4) dataset-specific fine-tuning for video captioning.

Training stage	Context	Compute	Duration
Image-to-video contrastive pre-training	Short	64 TPUv3	7 days
Short-to-long video contrastive pre-training	Long	256 TPUv3	1 day
Video-to-text tuning with 400M LM	Short	16 TPUv3	15 hours
Video-to-text tuning with 1B LM	Short	64 TPUv3	2 days
Video-to-text tuning with 400M LM	Long	128 TPUv3	2.5 days

Table 2. Compute resources and duration for training our video-first encoders and video-to-text models used to report results in Tables 2 and 3.

Each video corresponds to 10 different captions, so we report average scores.

- *EgoSchema*: We report results on the released subset with annotations (500 examples in total). We report results for the full set in Section D and discuss limitations. For reporting results on multiple-choice QA, a task that our model has not learned to perform, we follow [15] and InternVideo [23] and train our video-to-text model on MSRVT-TQA for 5k steps for adapting to the task.

C. Ablation Studies

C.1. Video–Language Pre-training

Here, we provide more insights on video–language pre-training, including data mixtures, model initialization, input context lengths, and auxiliary losses. For our ablations, we use ViT-Base/BERT-medium.

Image and video data mixtures. First, we present performance of our video model (i.e., joint space-time attention) when trained from scratch using different pre-training image and/or video datasets in Table 3. We report Recall@1 for zero-shot text-video retrieval and assess the quality of different pre-training datasets based on downstream perfor-

mance. Overall, the suitability of each dataset largely depends on the benchmarks. Some key observations in addition to our discussion in Section 6.3 can be summarized as follows:

- Dataset size alone was not a key factor for strong downstream performance (e.g., ALIGN with 1B examples (Line 1) vs. LTIP with 324M examples (Line 3) for image datasets, HowTo100M Clips with 100M examples (Line 6) vs. VTP with 27M examples (Line 5) for video).
- Domain match between train and inference time can be a catalyst for good performance, even in cases where vision–text alignment quality is poor (e.g., HowTo100M Clips for YouCook2 (Line 6, third column) in comparison with all other benchmarks). This becomes prominent in video datasets, which tend to be very domain-specific in comparison to image benchmarks.
- Training on videos is important for benchmarks with strong temporal dependencies, but not necessary for benchmarks that largely depend on spatial understanding. In particular, comparing LTIP and VTP which are of similar domain and quality (i.e. collected in a similar way) but of different sizes (324M vs 27M, respectively), we find that MSR-VTT is benefited more by a larger image dataset, whereas video-level information is more crucial than dataset size for VATEX. This further supports our observations from Section 6.3.
- Overall, image and video examples are complimentary for video understanding, across all benchmarks.

Two step image-then-video training. As discussed in Section 3.1, we first train an image ViT-based encoder on images, which we further tune on the video domain via joint space-time attention. This facilitates faster training and stronger contrastive objective due to larger batch sizes (i.e., 8k vs 512). We test the effect of the two-step training approach on zero-shot retrieval in Figure 1. We present % difference on video-to-text Recall@1 when (1) training

	Dataset Size	MSR-VTT		VATEX		YouCook2		ActivityNet	
		T2V	V2T	T2V	V2T	T2V	V2T	T2V	V2T
ALIGN [12]	1B	24.1	17.1	6.2	3.3	1.0	0.5	1.9	1.1
JFT [29]	300M	16.2	14.6	7.6	6.3	1.6	1.0	1.5	<u>3.2</u>
LTIP [1]	324M	27.4	21.0	10.3	5.3	2.7	1.3	3.4	2.4
All Image	1.6B	<u>31.1</u>	<u>27.7</u>	<u>18.6</u>	10.9	3.7	2.4	<u>4.8</u>	<u>3.2</u>
VTP [1]	27M	23.2	21.3	13.3	15.3	2.5	1.9	3.3	3.1
HowTo100M Clips [16]	100M	14.3	13.7	5.6	6.6	<u>9.6</u>	<u>9.9</u>	1.2	1.5
VideoCC3M [18]	7M	12.3	11.9	5.0	4.4	1.0	0.0	1.1	1.0
All Video	134M	26.2	24.7	14.9	<u>15.3</u>	7.7	6.5	4.3	3.0
All	1.8B	<u>36.3</u>	<u>33.8</u>	<u>20.6</u>	<u>23.3</u>	<u>9.3</u>	<u>9.6</u>	<u>6.2</u>	<u>5.1</u>

Table 3. Performance of video model per pre-training image and/or video dataset on zero-shot text-to-video (T2V) and video-to-text (V2T) retrieval (% Recall@1) when trained from scratch. Two best variants are underlined per benchmark.

	MSR-VTT		VATEX		YouCook2		ActivityNet	
	T2V	V2T	T2V	V2T	T2V	V2T	T2V	V2T
Joint ST-ViT (Contrastive only)	39.9	38.1	23.9	26.3	11.4	12.6	6.7	6.4
+ MLM loss	39.0	36.6	23.0	26.4	11.0	12.0	6.8	6.0
+ Captioning loss	39.9	36.0	22.0	24.3	11.2	11.5	6.8	5.6

Table 4. Text-video retrieval results (Recall@1) for different pre-training objectives. We apply 25% masking of the video input in all cases.

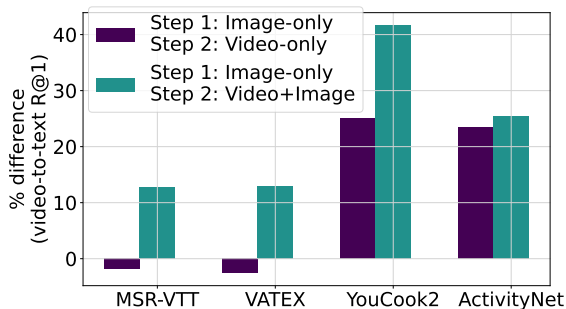


Figure 1. Performance difference (% video-to-text Recall@1) of two step image-then-video training approaches in contrast to training video models on image and video data from scratch.

first on image-only and then video-only, or (2) training first on image-only and then image+video, where the same image datasets are used again with smaller weights for gradient computation, in contrast to jointly train on image+video from scratch. For case (1) we see improvements for two out of four benchmarks. While previous work only continue pretraining on video data [2, 3, 25], we also test keeping image datasets in the pre-training mix (case 2), and surprisingly, we observe an even larger relative improvement across all benchmarks.

Given these observations we conclude that (a) there is indeed a benefit from pre-training first on images and then on videos, but (b) it is important to keep image samples in the mix, since video-text training is noisier and might have negative effect on spatial understanding, which can be mitigated in part from continual training on images. We confirm

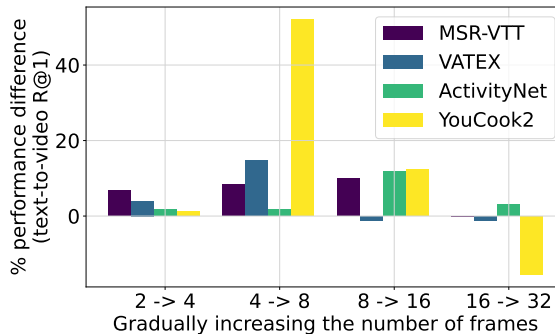


Figure 2. Performance difference (%) for zero-shot text-to-video Recall@1 when gradually increasing the number of frames. This indicates the sensitivity of each benchmark to the number of frames and hence acts as an indicator of the temporal dependencies present in different benchmarks that are widely used.

the latter by also directly evaluating on image benchmarks: for COCOCap [5] an image-trained ViT achieves 35% text-to-video Recall@1, whereas performance drops to 26% for case (1) and 29% for case (2).

Context length. We also experiment with variable number of frames per benchmark at a fixed FPS of 1. We range the number of frames from 2 up to 32 and present incremental performance difference as we gradually increase the number of frames in Figure 2. For most benchmarks, the performance improves as we increase the number of frames up to 16 frames, whereas longer inputs do not show benefits. This result highlights that most current benchmarks do not adequately measure temporal understanding (more than 16

frames). The most challenging dataset is YouCook2, where performance is close to random when considering less than 8 frames. VATEX also presents more challenging temporal dependencies with low performance when considering less than 4 frames. These observations further validate our key findings of Section 6.3 and shed some light on which academic benchmarks are more appropriate to use for evaluating video models.

Auxiliary losses. Although contrastive pre-training is a standard paradigm for image [12, 21] and video training [1, 14, 17, 24], prior work has explored captioning losses for training vision encoders in addition to or instead of contrastive objectives [8, 13, 22, 26, 28]. We next consider variants of the captioning loss for video pre-training as auxiliary losses to the contrastive objective with a 1:1 weighting between the two.

Adding a captioning loss in a dual encoder requires a multimodal encoder/decoder on top for fusing modalities and predicting tokens conditioned on the visual content. We add extra multimodal layers on top of the dual encoder similarly to [26]. We also consider two popular variants of the loss: (1) Masked Language Modeling (MLM), where we consider a multimodal *encoder* and mask 15% of the input textual tokens to predict, and (2) Captioning, where we consider a multimodal *decoder* instead and predict each token of the caption autoregressively. We present results for the different pre-training objectives in Table 4¹. Overall, neither variant is able to improve results; in contrast performance mostly drops by adding the extra objective. Our results contradict prior work’s observations on image pre-training [22] or video pre-training with frozen backbones [8, 13, 26, 28] and show that although contrastive objectives might be too coarse-grained for videos, considering captioning losses might be too fine-grained. We hypothesize that the very noisy video–text alignments hurt training of the video encoders when fully fine-tuned and the model needs to predict every textual token, which might not correspond to the visual input. In order to improve video pre-training in future work, we should look at either video-specific training objectives or better video–text alignments for video datasets.

C.2. Video-to-text Tuning

Synthetic VideoCC3M We compare the original and synthetic versions of VideoCC3M (see Section B) for video-to-text tuning in Figure 3. Given a version of SHORT-ViViT-L-to-text tuned only on image datasets from Table 3 and VTP for videos², we measure % performance difference

¹We apply 25% input masking for all variants for more efficient training.

²We exclude HowTo100M Clips for short video-to-text tuning, since the dataset is very noisy and leads to drastic drops in performance across

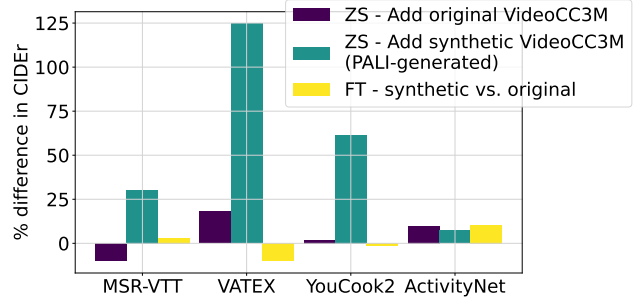


Figure 3. Performance difference (CIDEr) on zero-shot video captioning when we include either the original or synthetic VideoCC3M version (first two bars per benchmark). We also report performance difference between the two variants (i.e., synthetic vs. original dataset) when we fully fine-tune the models (third bar per benchmark).

on CIDEr for zero-shot video captioning when we include either version of VideoCC3M (two first bars per benchmark). We observe a very large performance improvement (up to 125% relative increase) when we use the synthetic version of VideoCC3M for three out of four benchmarks³. In contrast, using the original version of VideoCC3M provides moderate improvements for two out of four benchmarks (10-20%), no improvement for YouCook2 and has a negative effect on MSR-VTT.

However, observations do not hold when we fine-tune different model versions. In particular, we also report performance difference of the model trained with the synthetic dataset version against the one trained with the original one when we fine-tune them on the target datasets (third column per benchmark). In this case, we do not see benefit by using the synthetic dataset and even suffer performance drop for VATEX (10% relative decrease).

Overall, our findings for using synthetic video datasets are mixed. Our main hypothesis is that synthetic video captions can benefit model training in zero-shot settings since the model learns to produce *longer* and *more descriptive* captions (as empirically observed) which benefits metrics such as CIDEr. However, such benefits vanish when we further tune the model to the domain and style of interest.

Video-to-text Masking We mention in Section 6.1 that we additionally apply up to 30% masking for training and inference on video-to-text. In Figure 4, we present performance (Rouge-L) on video summarization on YouCook2 for the setting described in Section 6.1 when we apply masking at different ratios (10-20-30%) for three model variants: IMAGEViT, SHORTViViT, and LONGViViT (same models presented in Figure 5 of the main paper).

most benchmarks.

³Improvements for YouCook2 are smaller, i.e. less than 10%.

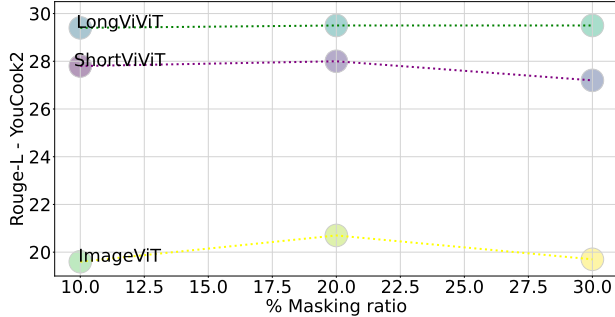


Figure 4. Rouge-L scores for long video models presented in Figure 5 of the main paper when applying different masking ratios during both training and inference ranging from 10 to 30%.

	ES-Subset	ES-Full
Inference with 256 frames		
IMAGEViT 1B	40.8	30.9
SHORTViViT 1B	47.9	31.0
LONGViViT 1B	56.8	33.3
Modular approaches with 16-frame video models		
SeViLA-to-SHORTViViT	49.6	31.3
IMAGEViT-to-Bard	35.0	35.0
SHORTViViT-to-Bard	42.0	36.2
SeViLA [27] 4B	25.7	22.7
PALI [6] 5B-to-Bard	44.8	39.2
Blind Bard	27.0	33.2
Previous SoTA [15, 23]	–	32.1

Table 5. EgoSchema results (% Accuracy on multiple-choice QA) for subset and full evaluation set.

Overall, we do not observe significant performance degradation when considering different masking ratios across all model variants, and hence we used 30% masking for our main experimental results on longer videos (Section 6.2).

D. Additional Experimental Results

EgoSchema full evaluation set. We present our results on both the subset and full set of EgoSchema in Table 5. Overall, we find that blind LLMs can answer a large percentage of questions without requiring any visual grounding (33.2% for the full set in contrast to 27.0% for the subset). Moreover, incorporating visual context to Bard via PALI captioning boosts performance only by 18% for the full set in comparison to 66% relative improvement for the full set. Hence, we find that models utilizing LLMs have an advantage for answering the full set questions and can achieve more competitive performance independently of the quality of the visual encodings. Finally, LONGViViT still achieves the best performance for the full set when compared with models employing LMs of equal size, and the

	PT
IMAGEViT 1B (32 frames @ 1FPS)	39.1
SHORTViViT 1B (32 frames @ 1FPS)	41.9
LONGViViT 1B (256 frames @ 5FPS)	45.7
IMAGEViT-to-Bard	37.8
SHORTViViT 1B-to-Bard	38.8
Flamingo [1] 3B	43.6
SeViLA [27] 4B	46.2
PALI [6] 5B-to-Bard	42.4
Blind Bard	36.8

Table 6. Accuracy (%) on multiple-choice QA on Perception Test (PT). Models in the second and third blocks process videos at 5 FPS, except for Flamingo and SeViLA which follow the settings reported on [19].

original SeViLA [27] model⁴ fails to address the task despite its size (*i.e.* 4B parameters). Given these observations, we overall find the subset to be more challenging than the full set for video understanding evaluation and invite future work to also report performance on the subset for complete comparisons.

Perception Test. We present results on multiple choice video-QA task of the recently released benchmark, Perception Test [19]. This benchmark is slightly out of the long video domain – videos are relatively short (<30 seconds), and run at 10 FPS. Nonetheless, we hypothesize that given the nature of questions involving actions and localization, models could benefit from higher FPS and subsequently, longer sequences of input frames. We present results for our model variants and modular methods in Table 6. Indeed, we are able to boost performance when processing videos at 5 FPS considering 256 input frames with LONGViViT (*i.e.* comparison of models in first block of Table 6). Our model still performs better than modular methods that utilize Bard, including PALI-3 for frame captioning. We moreover outperform reported zero-shot results by Flamingo-3B. Finally, we are very close (0.5% absolute difference) to SoTA SeViLA [27], a model with 4B parameters and separately tuned localizer, which however fails to generalize to longer videos and questions about longer-range dependencies (*i.e.* EgoSchema; Table 5).

E. Memory Consumption and Scaling

We present the peak memory consumption of our model versus a vanilla joint space-time video encoder when scaling in model size or input length in Figure 5. First, we can scale the size of our video encoder up to 6B parameters without requiring model sharding when using A100

⁴In the original setting, SeViLA Localizer is trained with 32 input frames uniformly sampled from the video and selects 4 frames to feed to the Answerer.

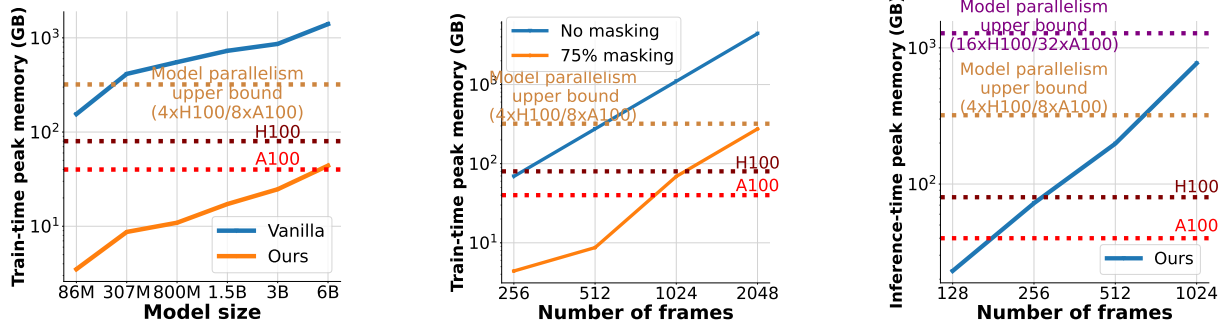


Figure 5. Peak train or inference time memory consumption with different model scales and number of input frames.

GPUs during training (first panel of Figure 5). Next, we can further scale the input sequence length to 2048 frames during training time when applying 75% token masking and to 1024 frames during inference considering a smaller percentage of masking (i.e., 30% for which we have noticed no significant performance drop) when sharding the model parameters across 8 A100 GPUs (second and third panels of Figure 5).

F. Examples

Finally, we present examples of the generated video summaries for the full length videos of YouCook2 in Table 7. For each example, we provide sparsely sampled frames from the entire video and the generated summaries by PALI-to-Bard, which is the most competitive model using LLM-assistance for information aggregation, and our LONGVIVIT model. As observed, PALI-to-Bard is able to provide very detailed descriptions of parts of the video, but it lacks consistency and coherence between the sentences of the output. Moreover, the model cannot focus on what is important in the video. For example, the main goal of the first video displayed in Table 7 is to demonstrate how to cook snails. However, the video includes several non-important clips, where people are discussing and/or change places from inside to outside and inside again. As a result, the summary provided by the model lacks coherence, describes isolated events and does not focus enough on the main point and ingredient of the video, which is the snails. In contrast, our model, although much smaller, is able to process the entire input sequence in one go and therefore provides more concise summaries, which remain coherent and do not include unimportant events. However, our model is still small-scale and does not employ an LLM for generation, which might lead to mistakes in the output. For example, the last video of Table 7 displays how to cook fried Schnitzel. In the generated summary by LONGVIVIT the frying of the chicken is repeated before and after adding the other ingredients (e.g., flour, salt, pepper), which is not true according to the video. We hypothesize that such mistakes can be eliminated when scaling the model size, and especially the LM component of our model.

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. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022. 2, 4, 5, 6
- [2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021. 1, 4
- [3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 4
- [4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 1
- [5] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 4
- [6] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023. 2, 6
- [7] R Christoph and Feichtenhofer Axel Pinz. Spatiotemporal residual networks for video action recognition. *Advances in neural information processing systems*, 2, 2016. 1
- [8] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. Violet: End-to-end video-language transformers with masked visual-token modeling. *arXiv preprint arXiv:2111.12681*, 2021. 5
- [9] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022. 2
- [10] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona




	
PALI-to-Bard	A man in a blue shirt is talking to another man. The man in the blue shirt is holding a piece of paper. They are talking about snails. The man in the blue shirt is holding a bowl of snails. He is putting the snails in a glass jar filled with water. He is cooking in a kitchen. He is picking herbs from a bunch of green plants. He is cooking food on a gas stove. He is cutting up a bunch of green vegetables on a white plate. He is putting a salad on a plate with a spoon.
LONGVIVIT	This is a video of cooking snails. They first use a knife to cut the snails. Then they put the snails in a pot and cook them. After that they put the snails in a bowl and put the sauce on them. Finally they eat the snails.
	
PALI-to-Bard	The video shows a chef preparing a dish called Van Rani Vab. The chef starts by mixing spices in a bowl. He then adds the spices to a pot of meat and sauce. He cooks the meat and sauce on the stove, stirring occasionally. Once the meat is cooked, the chef adds tomato sauce and saffron to the pot. He then cooks the dish for a few more minutes. Finally, the chef plates the dish and garnishes it with a mint leaf. The chef wears a red chef's jacket throughout the video. He is seen standing in front of a yellow wall, a microwave oven, and an oven. He also uses a variety of kitchen utensils, including a spoon, a fork, and a wooden spoon. The video ends with the chef eating the dish. He is seen holding a plate of food in front of a sign that says "Van".
LONGVIVIT	This is a video of cooking lamb shish kebab. They first cut the lamb into small pieces. Then they put the lamb in a pot and add the spices. After that they add the onions and cook it. Finally they add the water and cook it.
	
PALI-to-Bard	The video shows a person making Wiener Schnitzel. First, the person cracks an egg into a bowl and mixes it with a fork. Then, they pour flour on a piece of tin foil and coat a piece of meat in the flour. Next, they dip the meat in the egg mixture and then coat it in bread crumbs. Finally, they cook the meat in a frying pan and serve it with potatoes and sauce.
LONGVIVIT	This is a video of making fried chicken. They first put the chicken into the frying pan and cook it for four minutes. Then they add the flour, egg, salt, pepper, and the mayonnaise. After that they fry the chicken for four minutes. Finally they put the chicken on a plate and add the mayonnaise.

Table 7. Generated summaries of full-length YouCook2 videos by PALI-to-Bard and our LONGVIVIT model.

- Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 1
- [11] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021. 1
- [12] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 4, 5
- [13] Linjie Li, Zhe Gan, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Ce Liu, and Lijuan Wang. Lavender: Unifying video-language understanding as masked language modeling. *arXiv preprint arXiv:2206.07160*, 2022. 5
- [14] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 508:293–304, 2022. 5
- [15] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *arXiv preprint arXiv:2308.09126*, 2023. 3, 6
- [16] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 4
- [17] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 5
- [18] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XIV*, pages 407–426. Springer, 2022. 4
- [19] Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adria Recasens Contente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, et al. Perception test: A diagnostic benchmark for multimodal video models. *arXiv preprint arXiv:2305.13786*, 2023. 6
- [20] AJ Piergiovanni, Weicheng Kuo, and Anelia Angelova. Rethinking video vits: Sparse video tubes for joint image and video learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2214–2224, 2023. 1
- [21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry,

- Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5
- [22] Michael Tschannen, Manoj Kumar, Andreas Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. *arXiv preprint arXiv:2306.07915*, 2023. 5
- [23] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 3, 6
- [24] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6787–6800, 2021. 5
- [25] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3333–3343, 2022. 4
- [26] Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv preprint arXiv:2212.04979*, 2022. 2, 5
- [27] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. *arXiv preprint arXiv:2305.06988*, 2023. 6
- [28] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *Advances in Neural Information Processing Systems*, 34:23634–23651, 2021. 5
- [29] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022. 4