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

Pinelopi Papalampidi* Skanda Koppula* Shreya Pathak*
Justin Chiu Joe Heyward Viorica Patraucean Jiajun Shen Antoine Miech
Andrew Zisserman Aida Nematzdeh
Google DeepMind
{pinelopi,skandak,shreyapa}@google.com

Abstract

Understanding long, real-world videos requires modeling of long-range visual dependencies. To this end, we explore video-first architectures, building on the common paradigm of transferring large-scale, image–text models to video via shallow temporal fusion. However, we expose two limitations to the approach: (1) decreased spatial capabilities, likely due to poor video–language alignment in standard video datasets, and (2) higher memory consumption, bottlenecking the number of frames that can be processed. To mitigate the memory bottleneck, we systematically analyze the memory/accuracy trade-off of various efficient methods: factorized attention, parameter-efficient image-to-video adaptation, input masking, and multi-resolution patchification. Surprisingly, simply masking large portions of the video (up to 75%) during contrastive pre-training proves to be one of the most robust ways to scale encoders to videos up to 4.3 minutes at 1 FPS. Our simple approach for training long video-to-text models, which scales to 1B parameters, does not add new architectural complexity and is able to outperform the popular paradigm of using much larger LLMs as an information aggregator over segment-based information on benchmarks with long-range temporal dependencies (YouCook2, EgoSchema).

1. Introduction

Long-video understanding requires modeling of the temporal dynamics and long-range visual dependencies of real-world scenes [63, 64]. However, capturing long-range visual content is challenging, even when equipped with large language models. In this paper, we overcome hardware memory limitations and demonstrate how to extend video encoders to directly process minutes-long visual content using language grounding, and simple, established techniques without additional architectural complexity [24, 64]. We focus on long videos through the lens of language, assessing our models on the widely applicable tasks of visual summarization and question-answering.

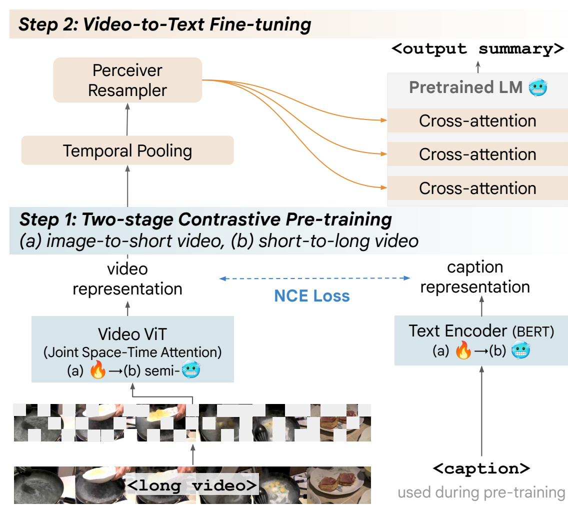

Figure 1. Two main training steps: (1) training a video encoder via Noise Contrastive Estimation and (2) using this frozen video encoder with a pre-trained, frozen LM and visual adapter layers for video-to-text generation (e.g., video summarization and Q/A).

Recent work on vision–language models have yielded impressive results, predominantly focusing on understanding images or short clips of 16 frames or less [1, 13, 30, 72, 73]. This work recycles strong pre-trained image encoders, performs late temporal fusion [1, 71, 73], and employs mostly-frozen, powerful LLMs. The lack of *video-first* encoders, equipped with early temporal aggregation, may handicap the ability to process complex visual dependencies, and this is usually reflected in prior work's focus on short video benchmarks ($<$ 30 seconds) in which sixteen frames are sufficient for competitive performance [4, 29].

In this work, we systematically explore video-first models starting from a standard image–language recipe using

*Equal contribution.

two-step training and large pre-trained LMs (Figure 1; [1]). This baseline enables us to start from a demonstrably scalable, simpler-to-tune, widely-used recipe that performs competitively [15, 30]. Through our analysis, we are able to scale this method in a memory-efficient manner to longer sequences of frames, up to 4.3 minutes of video at 1 FPS.

We first explore video-first models on short-video benchmarks (MSR-VTT [67], VATEX [60], YouCook2 [81], ActivityNet [28]) and compare against the SoTA VideoCoCa model [71]. We show that simple joint space-time attention significantly improves performance over frame-level encodings on benchmarks with rich temporal dependencies (YouCook2, VATEX). Overall, our models are able to reach VideoCoCa performance, while requiring fewer parameters and lower frame resolution.

This performance gain incurs extra compute and memory costs that grow quadratically with the video length. To address this, we provide one of the first systematic analyses of the memory/accuracy pareto-front of popular memory-efficient methods; this includes factorized attention, parameter-efficient image-to-video adaptation, input masking, and multi-resolution patchification. Through this analysis, we find that among all these options, simple token masking (up to 75%) during contrastive pre-training incurs only a 1% Recall@1 drop on zero-shot text-video retrieval, and no drop in zero-shot video captioning. At the same time, such high masking offers 2-3x memory savings and allows us to generalize to longer video contexts. The alternatives we explore (*e.g.*, efficient backbone architectures, more sophisticated TubeViT-style patchification [49]), do not maintain the same robustness against noisy video inputs and present a 25% relative decrease in performance for text-video retrieval on challenging benchmarks (YouCook2, VATEX). Finally, although parameter-efficient methods [21, 22] fail to adapt image encoders to video-first models without suffering performance drops, we find that they can adapt video models trained on short contexts (e.g., 16 second videos) to longer temporal horizons.

Based on the above learnings, we extend our best performing short-video encoder to longer contexts of 256 frames (4.3 minutes at 1 FPS). We use the full-length videos of HowTo100M [42] accompanied by LLM-generated summaries based on the ASR to further contrastively train our LONGVIVIT while masking 75% of the input video tokens and freezing most parameters of the encoder. LONGVIVIT-to-text ($\sim$1B parameters) is able to outperform modular methods that use LLM assistance and PALI-3 [9] for frame captioning on temporally rich benchmarks (YouCook2, EgoSchema). Even modular methods that consider frame selection (SeViLA [74]) or an oracle segmentation of the video for localizing and captioning key events (on YouCook2) cannot reach LONGVIVIT's performance. An interesting byproduct of our work is that we can glean which video–language benchmarks have strong temporal dependencies, and thus are suitable for testing long video models; we find that papers often use benchmarks in which short video or even blind models perform well [5, 41, 67].

In short, we provide the following contributions:

- We explore the memory/accuracy pareto-frontier of video-first vision–language models, and systematically evaluate many architectural, data, and training alternatives. In the end, we identify a simple recipe that enables scaling to 4.3 minutes at 1 FPS, many times longer than comparable video–language models [1, 71].
- We identify short and long video benchmarks with substantial temporal dependencies, for which we demonstrate that the traditional image-first, late-temporal fusion recipe is convincingly weaker than a video-first approach.
- Finally, we compare our long video models to a variety of strong baselines and show competitive performance with far fewer parameters; this includes baselines that use LLM-based aggregation over visual captions, and we quantitatively evaluate this common approach for the first time on standard video benchmarks.

2. Related Work

We base our recipes on [1, 30], which provide a strong two-step video–language recipe that leverages pre-trained LLMs and works at scale. Similar work at smaller scale has additionally included captioning losses [32, 76], more contrastive losses [10, 38, 43, 66], masking/masked autoencoding [15, 16, 18, 19, 33, 35, 40, 55], and combinations thereof [13, 23, 54, 58, 61, 72, 73, 75, 82]. This work focuses on image–text modeling and extends to $<$30 seconds via image-to-video transfer, selective fine-tuning, or temporal fusion of frame encodings [1, 71, 73].

A volume of work focuses on video-first learning. This includes some of the very early work in image-to-video kernel inflation [6, 53, 56], transformer-based video architectures [2, 3, 37], image-to-video parameter-efficient adaption [7, 36, 46], and multiple spatiotemporal resolutions along different network paths [14, 39, 68, 70]. These have still only been demonstrated on short videos, so other works have broached the challenge of temporal scalability: [24, 51, 64] propose alternative encoders, and [27, 48, 59] propose more exotic attention mechanisms. TubeViT [49] proposes multi-granularity patchification. We systematically dissect what works and scales among some of these alternatives, electing options that enable us to re-use strong pre-trained models and use standard, more easily-tuned architectures.

Specifically in video-to-text generation, approaches that handle longer videos are very limited and mostly target images or short videos [15, 31, 61]. A dominant approach is to summarize frames and aggregate information via LLMs [31, 34, 62, 77]. To the best of our knowledge, we are the first to attempt to train large-scale video-to-text models on longer sequences of frames and directly test them against LLM-assisted modular methods on challenging temporal benchmarks [41, 81].

3. The Video-to-Text Architecture

We base our approach on the successful two-step recipe that combines pre-trained vision and language models [*e.g.*, 1, 30, 72, 73] as shown in Figure 1: (1) we first pre-train a vision encoder, and then (2) fuse the frozen vision representations into a pre-trained, frozen LM.

3.1. Video–Language Contrastive Pre-training

Following common practice [1, 30], we use a dual vision–language architecture with a Noise Contrastive Estimation (NCE) loss [17, 45, 65] to pre-train our vision encoder, similar to CLIP [50], ALIGN [26] and VideoCLIP [66]. Both encoders are transformers [57]: a BERT-medium (77M) or base (117M) language encoder and ViT-Base (86M parameters) or Large (307M parameters) vision encoder. On the language side, caption representations are computed by averaging across the corresponding token representations. On the vision side, video frames are patchified into a sequence of visual tokens, fed into a vision encoder, and then average pooled to produce a final video representation.

Most prior larger-scale video–language models use pre-trained image encoders and patchify frames individually via 2D convolutions [*e.g.*, 1, 66, 71]. Instead, we create spatiotemporal tubelets via 3D convolutions as done in recent vision-only models [2, 49, 55]. Using 3D tubelets instead of flat patches has the dual advantage of higher input compression and more explicit temporal contextualization; our early experiments yielded improved performance. The tubelet embedding sequence is then flattened, added to learnable positional embeddings, and fed into the vision encoder. The vision encoder uses spatio-temporal attention as in ViViT [2]: *Joint space-time attention* does not add any new parameters to vanilla image ViT [12], facilitating transfer between image and video models.

Training a large-scale transformer-based video encoder can be challenging because self-attention across thousands of visual tokens is both compute and memory intensive. Memory bottlenecks a model in two ways: (1) limiting the number of frames, and (2) limiting the contrastive batch size during training, negatively impacting performance. To address (2), we use a pre-trained image encoder trained with large batch sizes, and further tune it on videos, instead of jointly training from scratch on images and videos. For initializing the 3D convolution, we repeat the pre-trained weights across the temporal dimension similarly to [2] (see Appendix A). During video–language pre-training, we maintain different embedding paths for images vs. videos: images are embedded with the original 2D convolution and videos with a separate 3D convolution (no weight sharing).

3.2. Video-to-Text Tuning

We follow prior work [*e.g.*, 1, 72, 73] by plugging the frozen pre-trained vision encoder into a frozen pre-trained LM. We first temporally mean pool the video representations to keep a fixed number of tokens independently of the number of frames and next use a randomly initialized Perceiver-resampler [25] to project the representations to the LM embedding space (Appendix A). We add new randomly initialized cross-attention layers at each layer of the LM to ground generation on the visual content. We train the new layers and Perceiver resampler with a standard auto-regressive video captioning loss: $-\log p(w_t|w < t; \mathcal{V})$, where w_t is its t^{th} token, and $\mathcal{V}$ is the video representation.

4. Memory-Efficient Encoder Design Space

Device memory is a key bottleneck for video training with joint space-time attention. To overcome this, we explore four broad categories of solutions: (1) efficient attention, (2) parameter-efficient image-to-video adaptation, (3) input token masking, and (4) multi-resolution patchification.

1. Attention mechanism. Factorized attention [2, 3] separates the temporal and spatial dimensions over which self-attention is applied, reducing both memory and computational costs. However, this modification introduces a new temporal block within each transformer layer making initialization and model tuning more challenging. In contrast to [2], that initializes the new blocks with zeroes, we find that we achieve best performance when initializing the temporal blocks with the same self-attention weights of ViT. However, we add a gating mechanism which acts as a residual connection between the self-attention blocks: $h = h + tanh(\alpha)h_{temporal}$. Here, α is a trainable parameter initialized to zero, that helps maintain the capabilities of the original ViT during training.

2. Parameter-efficient adaptation. We explore using parameter-efficient methods from NLP [8] to adapt image encoders to video, while only tuning a small percentage of model parameters. Most prior work adapts image-based models by freezing an image backbone and adding late, trainable temporal-fusion layers [10, 71, 78]. In contrast, we explore ways to use pre-trained image encoders and adapt them to *video-first* architectures [7, 36, 46]. Inspired by the success of parameter-efficient adaptation in NLP [79], we consider using MLP Adapters [21] and LoRA [22] (details in Appendix A). We also explore tuning only temporal self-attention blocks [7], effectively as adapter layers, in factorized attention. In all variants, we still tune the video-specific 3D patch convolution.

3. Token masking. Most existing work samples videos at a fixed frames per second (FPS) rate [*e.g.*, 1, 2, 55, 74]. However, semantics required for many video–language tasks vary slowly in the temporal dimension [80] and videos present high degree of redundancy between consecutive frames [55]. We explore ways to sparsely sample the video input to reduce the number of input visual tokens. Specifically, we test random masking of input tubelet embeddings. Since consecutive frames are largely redundant, the same semantic signals could potentially be extracted even with high masking rates. For example, [55] masks up to 95% of the input video to reach optimal performance on the task of video-masked autoencoding. We demonstrate similar results in a video–language setting.

4. Multi-resolution patchification. Finally, we test a simple approach to reduce redundancy in videos via more coarse-grained patchification in the temporal or spatial dimension, as commonly done in multiple-view video models [14, 39, 70]. However, this decreases frame resolution, and may lose fine-grained information. As a result, we also experiment with TubeViT [49] variant that combines flat patches and tubelets of different granularity to mitigate information loss. Following [49], we use four different convolution kernels that can encode either coarse-grained temporal or spatial information; details are in Appendix A.

5. Datasets and Benchmarks

For contrastive pre-training, we use: (1) 27M video-text pairs (VTP) as described in [1], (2) HowTo100M [42] (HT100M; 100M instructional YouTube clips aligned with ASR using their timestamps, called HowTo100M Clips), and (3) VideoCC3M [44] (3M video-text pairs based on Conceptual Captions [52]). Unfortunately, we find the text–video alignment in VideoCC3M to be of poor quality; instead, we use a modified variant with generated pseudo-labeled captions of every video by PALI [9] (see Appendices B, C). To pre-train with longer videos, we use a long version of HowTo100M (referred to as HowTo100M Summary) consisting of (1) the full-length videos with an average duration of 6.5 minutes and (2) their textual summaries generated by automatically cleaning and summarizing the ASR transcripts using an LLM [20]. We also include the image datasets of [1]. For video-to-text tuning, we use the same mixture of datasets but exclude HowTo100M Clips, since the noisy video-text alignments hurt performance.

We report text-video retrieval and captioning results on *short video benchmarks*, with average video length $\leq$30 seconds: MSR-VTT [67], YouCook2 [81], ActivityNet Captions [28], and VATEX [60]. To evaluate performance on longer videos, we consider video summarization on full-length versions of YouCook2 and ActivityNet Captions, with a video duration of up to 5 minutes, and multiple-choice video question answering (QA) on EgoSchema [41].

6. Experimental Results

In Section 6.1, we describe our results evaluating alternatives in memory-efficient video encoder design; options described in Section 4. For this analysis, we use ViT-B/BERT-medium, with training details in Appendix B and ablations on experimental design in Appendix C.

	MSR-VTT		VATEX		YC2		AN	
	T2V	V2T	T2V	V2T	T2V	V2T	T2V	V2T
Joint ST-ViViT	39.6	**38.1**	23.8	**26.3**	**12.3**	**13.6**	6.7	6.4
Factorized ST-ViViT	**40.2**	36.9	**25.3**	25.4	11.6	12.7	6.6	**7.4**
Avg Frame-level	39.3	34.8	24.8	25.0	9.1	7.9	**6.8**	7.1
Att-pool Frame-level	38.4	37.5	21.9	26.1	9.0	8.9	6.1	6.2

Table 1. Text-video retrieval results (% Recall@1) when considering different visual backbones.

In Section 6.2, we combine our most competitive design choices from 6.1 and test our models on short and long video understanding benchmarks. We scale our best model variants to ViT-L/BERT-base with a 400M (or 1B) language decoder. We test our short video models on text-video retrieval and video captioning, and our long video models on video summarization and QA on 256-frame videos.

In Section 6.3, we share our experience working across short and long video benchmarks [5, 11, 41, 60, 67], offering insights about which ones yield robust temporal signal.

6.1. Exploration of Memory-Efficient Designs

We explore memory-efficient methods to train video-first encoders as described in Section 4. We first consider short video inputs of 16 frames at 1 FPS and report peak train-time memory consumption vs. performance on text-video retrieval on short video benchmarks [5]. Then, we test whether our main findings hold for longer inputs (128+ frames) on video summarization on full-length YouCook2.

Base architectures. We explore the memory/accuracy trade-off of different visual backbones in Table 1: ViViT with joint space-time attention (*i.e*., Joint ST-ViViT), ViViT with factorized attention (*i.e*., Factorized ST-ViViT) [2], and frame-level (ViT-based) image encodings with average or attentional pooling ('att-pool') [1, 71]. Different methods perform similarly, especially on MSR-VTT and ActivityNet (AN). Interestingly, attentional pooling on top of frame-level encodings does not improve performance. ViViT with either joint or factorized attention performs best and presents higher gains for YouCook2 (YC2), the more temporally challenging benchmark [6.3]. In contrast to prior work [*e.g*., 10, 71] which tests frozen image-to-video transfer and claims joint attention to be inferior, we find it to be competitive in this fully fine-tuned setting.

Architectures and token masking. We now test robustness of backbones when masking part of the input tubelets (0-75%). We report Recall@1 on text-to-video retrieval for YouCook2 and VATEX[1] per backbone for different masking

[1]We do not observe significant sensitivity to input masking for MSR-VTT and ActivityNet Captions across all configurations (Section 6.3).

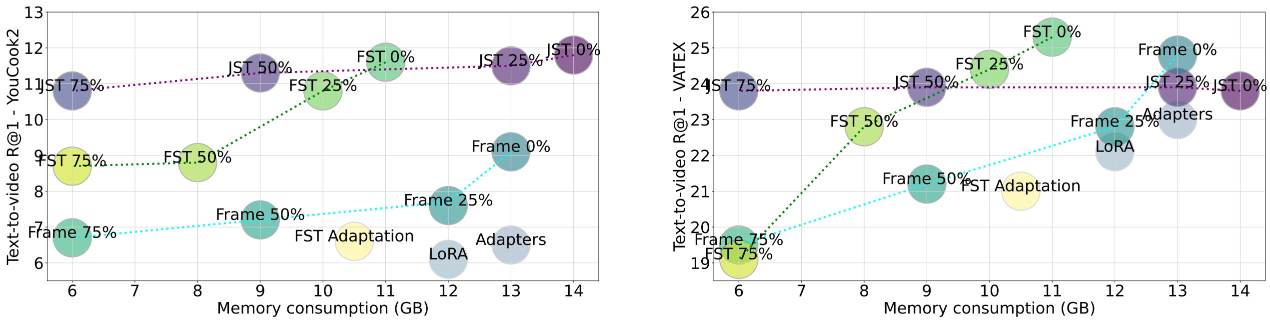

Figure 2. Trade-offs between performance (% text-to-video Recall@1; y axis) and train-time memory consumption (x axis) for different backbones (joint space-time (JST), factorized space-time (FST), and drame-level encodings) with random input masking (0% up to 75%) or parameter-efficient methods for training (Adapters, LoRA, factorized temporal (FST) adaptation; lower opacity).

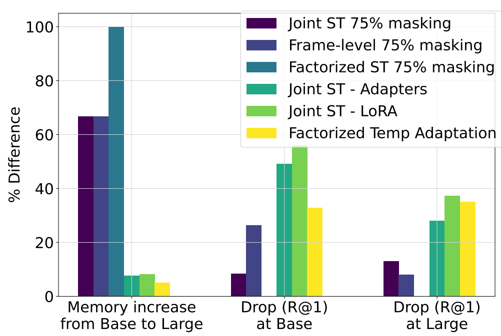

Figure 3. Difference (%) in memory consumption for different model scales: (ViT-B vs ViT-L). We also report performance drop of efficient methods presented in Figure 2 in comparison with the vanilla approach (i.e., no input masking and full fine-tuning) at different model scales to test whether behavior is similar.

ratios in Figure 2. Joint space-time attention (JST) is robust against noise from masking up to 75% during pre-training. The same *does not* hold for frame-level encodings and factorized attention (FST), where performance drops consistently as we increase masking. We conclude that JST can better handle noisy inputs and use it in further exploration.

Parameter-efficient adaptation. We next report performance of parameter-efficient image-to-video adaptation in Figure 2. We consider (1) JST with (a) MLP *Adapters* at every layer of the encoder, (b) *LoRA* with rank decomposition matrices in the self-attention and feed-forward transformer blocks, and (2) *factorized temporal adaptation* where we tune the temporal self-attention. No adaptation method can reach the memory savings provided by high input masking, since we tune parameters *depthwise* and gradient computation still requires backpropagation through the model. At the same time, we see significant performance drop, suggesting that adaptation of spatial-only models to the temporal dimension cannot be sufficiently addressed in semi-frozen fashion. Comparing parameter-efficient methods, we find MLP Adapters to be more competitive than LoRA, which is now canonical for LLMs. We hypothesize that LoRA is successful for tuning very small portions of the network and performing "easier" in-modality transfer.

Adaptation at scale. We next scale from ViT-B/86M to ViT-L/307M in Figure 3 and test whether observations hold with different model scales. We present the % memory increase from base to large (left bar set) and % performance *decrease* of each method at each scale[2]. Joint ST exhibits a similar memory pattern to frame-level, while leading to smaller accuracy drops, whereas factorized ST presents significant memory overhead with model scale due to the extra temporal parameters. For this reason, we exclude factorized ST from further experimentation. Finally, parameter-efficient methods are unable to achieve competitive performance at both model scales, although their memory requirements scale better with model size.

Multi-resolution patchification. Given the outsized memory impact of input token count in Figure 4, we additionally analyze: (1) *coarse-grained patchification* in the temporal (convolution over 4 instead of 2 frames) and/or spatial (convolution over 32x32 instead of 16x16 pixel spaces) dimension, and (2) the *TubeViT* [49] approach of multiple tube kernels of different spatiotemporal size and strides. For all benchmarks, masking the input at high ratios while maintaining a fine granularity of tubelets decreases performance significantly less than other input processing methods. Temporal coarse-grained patchification negatively affects benchmarks with richer temporal dependencies (i.e., YouCook2, VATEX) more than spatial. The opposite trend holds for datasets depending on spatial understanding (i.e., MSR-VTT, ActivityNet Captions[3]). TubeViT acts as the middle ground between the two by employing multiple kernels, with some performance degradation across all benchmarks. However, it is not able to alleviate the negative effects caused by considering coarser-

[2]Performance drop for factorized ST is omitted since the variant without masking leads to out of memory issues.

[3]Omitted from Figure 4 but follows same patterns as MSR-VTT.

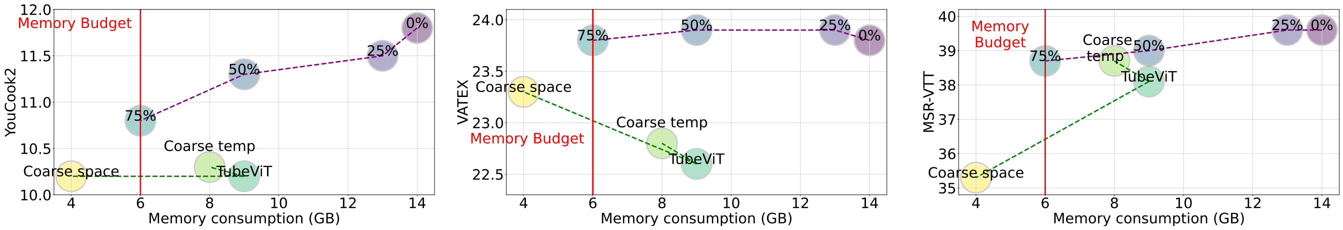

Figure 4. Trade-offs between performance (text-to-video Recall@1; y axis) and memory consumption (x axis) for input sampling methods: (1) high input masking ratios (0% to 75%) with joint space-time attention, (2) coarse-grained temporal (Coarse temp) and/or spatial (Coarse space) patchification with a fixed kernel and TubeViT which samples parts of the video with multiple 3D kernels of different granularity.

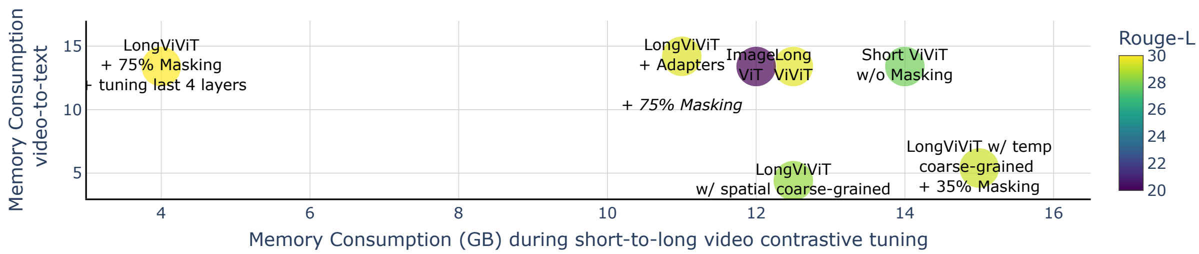

Figure 5. Scaling memory-efficient methods to more frames (i.e., 128 frames) for ViViT-B and variants. We measure performance for video-to-text summarization on the full-length YouCook2 videos via Rouge-L (color-coded) while keeping track of memory consumption during short-to-long video contrastive tuning (x-axis) and video-to-text tuning (y-axis).

grained information and presents higher memory requirements due to the multiple convolutions. Overall, we find that high masking with Joint ST and small tubelets yields the strongest memory/performance curves.

Scaling to longer videos. We now test the best methods from Figure 4 on 128 input frames (32.7k visual tokens). We select methods that are within a memory budget (red vertical lines) and would fit on a 16GB device when expanded to long videos (128+ frames). We contrastively fine-tune [3.1] our best performing video model (i.e., Joint ST referred to as SHORTVIVIT) on sequences of 128 frames on HowTo100M Summary [5], as detailed in Appendix B. We refer to this model as LONGVIVIT. Finally, we fine-tune LONGVIVIT for text generation (Section 3.2) on the full-length YouCook2, and report Rouge-L in Figure 5, measuring memory consumption during both long-context contrastive (x-axis) and video-to-text (y-axis) tuning.

Validating our previous results, IMAGEVIT (frame-level encodings) trained on longer videos with 75% masking[4] significantly under-performs video-first models (10 R-L drop). SHORTVIVIT without further HT100M Summary training performs better than IMAGEVIT, but cannot match models adapted to longer videos. LONGVIVIT improves performance by 1.8 Rouge-L points over SHORTVIVIT. Comparing input masking with coarser-grained patchification[5] provides similar insights to the previous paragraph.

[4]We start from IMAGEVIT trained on short videos with no masking.

[5]Using the same fine-grained SHORTVIVIT model for initialization.

Finally, we test MLP Adapters [21] for tuning SHORTVIVIT to longer videos and observe no performance drop in comparison with full fine-tuning. This provides further evidence that parameter-efficient methods can be used for "easier transfers" but not temporal adaptation of spatial-only models. One downside of MLP Adapters is that it increases parameter count during video-to-text tuning (y-axis in Figure 5). Thus, we also experiment with contrastively tuning only the last four layers of the model. With this, we observe a further 3x decrease in memory, since we tune the network *widthwise* and excise early layer gradient computation. At the same time, there is no memory increase for video-to-text and no performance degradation. We conclude that this combination (high input masking and tuning the last layers) is an effective setting for longer video adaptation. Given the observed robustness to masking, to further decrease video-to-text memory, we also mask 30% of the input video during training and inference without observing any drop in summarization performance (see Appendix C).

6.2. Main Results

Short video benchmarks. We present our main results on short video benchmarks in Table 2. We use ViT-L with BERT-base for contrastive pre-training (Section 3.1) and a 400M frozen LM for video-to-text tuning (Section 3.2). Our entire video-to-text model accounts for ~900M parameters, although we additionally test scaling the frozen LM to 1B parameters (~1.5B total count). We report Recall@1 for zero-shot text-video retrieval and CIDEr for zero-shot and

	MSR-VTT Zero-shot		FT	VATEX Zero-shot		FT	YouCook2 Zero-shot		FT	ActivityNet Zero-shot		FT
	T2V/V2T	C1/C2	C1	T2V/V2T	C1/C2	C1	T2V/V2T	C1/C2	C1	T2V/V2T	C1/C2	C1
ImageViT-L	30.9/**41.6**	24.6/25.1	63.6	36.2/**42.9**	37.9/39.4	61.1	18.2/16.8	14.5/16.5	95.9	20.6/18.2	16.3/17.7	41.1
ShortViViT-L	31.9/38.9	32.7/32.9	63.1	**37.8/42.8**	**43.6**/43.0	**67.5**	**20.4/20.5**	**21.0**/22.1	**131.9**	**21.3/18.9**	25.2/26.1	**44.8**
EffShortViViT-L	29.9/38.3	**33.8/33.9**	**63.8**	34.4/42.7	41.3/42.7	64.7	**20.5/20.3**	**21.1**/21.7	127.1	20.1/17.7	**27.0**/26.5	41.1
VideoCoCa-L [71]	**33.3**/–	24.3	–	–	–	–	18.9/–	20.7	–	31.5*/–	17.4	–
VideoCoCa-2.1B	34.3/64.7	27.1	73.2	53.2/73.6	22.8	77.8	20.3/–	34.3	128.0	34.5*/33.0*	19.3	39.3
Flamingo-3B [1]	–	–	–	–	40.1	–	–	55.8	–	–	–	–

Table 2. We present three model variants: ImageViT-L, that uses frame-level encodings with a late temporal fusion trained on images and videos, ShortViViT-L, our best performing video-first model with joint space-time attention, and Efficient ShortViViT-L (EffShortViViT-L) where we apply 75% train-time masking for 3x memory savings. We also report performance for SoTA image-first models: VideoCoCa-L and Flamingo-3B, although they are bigger and not directly comparable. We report Recall@1 for zero-shot text-to-video (T2V) and video-to-text (V2T) retrieval, and CIDEr for zero-shot and fine-tuned (FT) captioning when considering a 400M (C1) or 1B (C2) frozen LM for generation. ActivityNet retrieval results marked with * are not directly comparable, as these models uniformly sample frames, whereas we use the first frames of the long video with a fixed FPS of 1 to match experimental settings across benchmarks.

fine-tuned video captioning. We consider three model variants: frame-level encodings ImageViT, ShortViViT, and ShortViViT with 75% masking that uses 2-3x less memory (referred to as *Efficient* ShortViViT). We also report results for VideoCoCa [71] and Flamingo [1][6].

Our results remain consistent with our earlier observations. Contextualizing only intra-frame dependencies coupled with late temporal fusion (ImageViT) leads to inferior performance for retrieval and captioning on benchmarks with richer temporal dependencies (YouCook2, VATEX) but performs better on retrieval on MSR-VTT which relies on spatial understanding. Video-first architectures further tuned on video datasets (substantially noisier than curated image ones) improve temporal capabilities at the expense of spatial. For Efficient ShortViViT, we find that masking 75% of the input video causes a performance drop: an average of 1% absolute difference on zero-shot retrieval and no significant difference on zero-shot captioning across all benchmarks. The efficient model still performs similarly or better than ImageViT, especially on captioning and temporally rich benchmarks (e.g., YouCook2, VATEX), while consuming significantly less memory. Finally, when scaling the frozen LM component from 400M to 1B (C1→C2) for zero-shot video-to-text generation, we observe moderate improvements across benchmarks and variants.

We compare our results against large image-based models with SoTA performance on video benchmarks (second block of Table 2). Although results are not directly comparable due to different experimental settings, we are competitive and achieve even better results for temporally rich benchmarks (i.e., YouCook2) on text-video retrieval for models of similar parameter count[7]. Moreover, our models significantly outperform VideoCoCa on most video captioning benchmarks even when considering their much larger versions in the zero-shot setting. Finally, when fine-tuning our video-to-text models with the 400M LM, we are again able to match and surpass the performance of the larger VideoCoCa-2.1B in two out of four benchmarks.

Long video understanding. We further tune LongViViT-L on 256-frame HT100M Summary videos and evaluate zero-shot/fine-tuned summarization (YouCook2, ActivityNet) and QA (EgoSchema released subset); this is shown in Table 3. We additionally report results of LongViViT on Perception Test [47] in Appendix D, where videos are short but can benefit from higher FPS.

We consider two families of models. 1. Models that take as input 256 frames (first block of Table 3): ImageViT and ShortViViT pre-trained on 16-frame clips, and LongViViT further trained on 256-frame clips. 2. Modular approaches from prior work (second block of Table 3): (a) SeViLA Localizer [74] for localizing important frames in the long video given a textual query which are then fed into ShortViViT for performing the task[8], and (b) the popular paradigm of captioning video segments or frames and using an LLM to aggregate information and form coherent summaries or answer questions [31, 34, 77]. We try the latter approach with ImageViT and ShortViViT, generating captions over 16-second video segments and then feeding the captions to the September 2023 release of Bard, a much larger LLM than the ones used in previous results. We caption clips using uniform video segmentation (every 16 seconds) or an oracle segmentation when available (i.e., we consider ground-truth start and end timestamps for different events within ActivityNet and YouCook2 videos). We

[6]Models are not directly comparable due to different pre-training datasets, model sizes, training regimes, and input resolution. For instance, [71] fully fine-tune the LM and report results for 576 × 576 frame resolution instead of 256 × 256.

[7]Video-text retrieval results on ActivityNet Captions are not comparable since we are only considering the first 16 seconds of the video, whereas [71] uniformly sample frames from the entire video (∼180 seconds).

[8]We select 16 frames using the pre-trained localizer provided by [74]. For video summarization, we use synthetic summaries of the video generated by PALI+Bard as the textual query for retrieving frames.

	Zero-shot			Fine-tuned	
	AN	YC2	ES	AN	YC2
Inference with 256 frames					
ImageViT	14.4	4.6	40.8	23.8	29.4
ShortViViT	15.4	7.0	47.9	**24.3**	29.5
LongViViT	15.2	**20.3**	**56.8**	24.0	**30.6**
Modular approaches with 16-frame video models					
SeViLA-to-ShortViViT	16.2	4.2	49.6	**24.4**	28.3
ImageViT-to-Bard	18.1	15.8	35.0	22.9	19.1
+ oracle segments	16.3	16.2	–	22.7	22.1
ShortViViT-to-Bard	19.3	18.1	42.0	22.7	20.8
+ oracle segments	18.3	18.2	–	22.7	24.7
PALI [9] 5B-to-Bard	**22.0**	19.9	44.8	–	–
Blind Bard	–	–	27.0	–	–
SoTA [69]	–	–	–	*36.9*	*34.6*

Table 3. Results on long video-to-text benchmarks. We report Rouge-L for zero-shot and fine-tuned video summarization on ActivityNet Captions (AN) and YouCook2 (YC2) and zero-shot accuracy (%) for multiple choice QA on EgoSchema (ES).

also test substituting our small video models with PALI-3 (5B parameters) for frame captioning[9]. Finally, we reference the SoTA fine-tuned performance on ActivityNet and YouCook2, when using specialized models with pre-computed features by multiple networks, object detectors, and domain-specific vocabulary [69].

Looking through Table 3, we find that on ActivityNet, which contains less temporal dependencies [6.3], modular approaches via frame selection or LLM-based aggregation of information (second block) perform well. Frame captioning via PALI combined with the power of LLMs is enough for the task in a zero-shot setting. For fine-tuned models, feeding either the long input or selected frames into ShortViViT perform better than utilizing Bard. On ActivityNet, we see no benefit from training further on longer videos.

In contrast, we find that short video and modular models are insufficient for addressing video tasks with longer-range temporal dependencies (YouCook2, EgoSchema). Adapting ShortViViT to longer contexts (LongViViT) significantly improves performance and achieves the best scores across all comparison approaches. Using Bard as an information aggregator over individual clip captions cannot achieve competitive performance, even when considering an oracle video segmentation for YouCook2 (Lines 3 and 5 in the second block of Table 3). Surprisingly, even using a much larger and more powerful image-based model (PALI) cannot reach LongViViT on YouCook2 and EgoSchema. Interestingly, selecting 16 key frames and feeding them into ShortViViT also outperforms Bard-based methods on EgoSchema and fine-tuned YouCook2. This suggests there can be temporal dependencies in long videos that cannot be resolved even with an optimal event segmentation for the video, or be aggregated by LLMs given inprecise visual information. On such benchmarks, LongViViT demonstrates strong performance even without LLM assistance.

[9]We consider captions of key frames per 8 seconds of video.

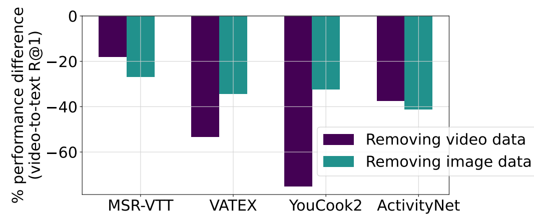

Figure 6. Performance difference (%) per benchmark when we remove (1) video or (2) image data from the training mixture.

6.3. Brief Notes on Video Evaluations

We briefly describe some of our findings on video evaluations. Firstly, we find that blind Bard is able to achieve SoTA results on the *full set* of EgoSchema (no visual input; 33.9% accuracy vs. 32.1% for the best model in [41]). Adding visual information from PALI into Bard increases performance to just 39.2%. However, on EgoSchema's released *subset*, performance of blind Bard is 27%, which is much lower than PALI-to-Bard (44.8%), suggesting that the subset contains questions that rely more on visual grounding than pure language reasoning, so we report numbers on the subset in Table 3 and on the full set in Appendix **??**.

Figure 6 details a simple ablation across other video benchmarks to quantify temporal richness. We test removing either video or image data from the training mix and measure the effect on performance (video-to-text Recall@1). We see a dramatic performance drop when removing video data for YouCook2 and VATEX (up to 75%). ActivityNet and MSRVTT suffer more from the absence of image data, whereas non-video training influences performance in lesser degree (as little as 18% for MSR-VTT). We believe there's room for more fine-grained, temporal-focused video–language benchmarks in the community.

7. Conclusions

In short, we systematically analyze memory-efficient methods to scale video-first architectures to longer sequences of frames and demonstrate that just masking high percentages of the video ($\leq$75%) yields competitive results on long video–language tasks. Such masking shows a very small performance drop on short videos, provides 2-3x memory savings and allows scaling up to 4.3 minutes at 1 FPS (LongViViT) when freezing part of the short video network in our two-stage training. LongViViT outperforms modular approaches with LLM assistance on video summarization and QA on benchmarks with richer temporal dependencies (YouCook2, EgoSchema). We overall demonstrate that encoding longer-range visual dependencies can make a difference in downstream performance and corrects mistakes that LLMs are unable to rectify.

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. 1, 2, 3, 4, 7

[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. 2, 3, 4

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, page 4, 2021. 2, 3

[4] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the" video" in video-language understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2917–2927, 2022. 1

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 2, 4

[6] 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. 2

[7] Dongsheng Chen, Chaofan Tao, Lu Hou, Lifeng Shang, Xin Jiang, and Qun Liu. Litevl: Efficient video-language learning with enhanced spatial-temporal modeling. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7985–7997, 2022. 2, 3

[8] Jiaao Chen, Aston Zhang, Xingjian Shi, Mu Li, Alex Smola, and Diyi Yang. Parameter-efficient fine-tuning design spaces. *arXiv preprint arXiv:2301.01821*, 2023. 3

[9] 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, 4, 8

[10] Feng Cheng, Xizi Wang, Jie Lei, David Crandall, Mohit Bansal, and Gedas Bertasius. Vindlu: A recipe for effective video-and-language pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10739–10750, 2023. 2, 3, 4

[11] Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2634–2641, 2013. 4

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3

[13] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 1, 2

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 2, 4

[15] 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. 2

[16] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. An empirical study of end-to-end video-language transformers with masked visual modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22898–22909, 2023. 2

[17] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010. 3

[18] Tengda Han, Weidi Xie, and Andrew Zisserman. Turbo training with token dropout. *arXiv preprint arXiv:2210.04889*, 2022. 2

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Doll'ar, and Ross B Girshick. Masked autoencoders are scalable vision learners. 2022 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15979–15988, 2021. 2

[20] 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. 4

[21] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR, 2019. 2, 3, 6

[22] 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. 2, 3

[23] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Qiang Liu, et al. Language is not all you need: Aligning perception with language models. *arXiv preprint arXiv:2302.14045*, 2023. 2

[24] Md Mohaiminul Islam and Gedas Bertasius. Long movie clip classification with state-space video models. In *European Conference on Computer Vision*, pages 87–104. Springer, 2022. 1, 2

[25] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021. 3

[26] 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. 3

[27] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020. 2

[28] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715, 2017. 2, 4

[29] Jie Lei, Tamara L Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. *arXiv preprint arXiv:2206.03428*, 2022. 1

[30] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 1, 2, 3

[31] KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 2, 7

[32] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23119–23129, 2023. 2

[33] Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. Scaling language-image pre-training via masking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23390–23400, 2023. 2

[34] Kevin Lin, Faisal Ahmed, Linjie Li, Chung-Ching Lin, Ehsan Azarnasab, Zhengyuan Yang, Jianfeng Wang, Lin Liang, Zicheng Liu, Yumao Lu, Ce Liu, and Lijuan Wang. Mm-vid: Advancing video understanding with gpt-4v(ision), 2023. 2, 7

[35] Yuanze Lin, Chen Wei, Huiyu Wang, Alan Yuille, and Cihang Xie. Smaug: Sparse masked autoencoder for efficient video-language pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2459–2469, 2023. 2

[36] Ruyang Liu, Jingjia Huang, Ge Li, Jiashi Feng, Xinglong Wu, and Thomas H Li. Revisiting temporal modeling for clip-based image-to-video knowledge transferring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6555–6564, 2023. 2, 3

[37] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022. 2

[38] 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. 2

[39] Chuofan Ma, Qiushan Guo, Yi Jiang, Ping Luo, Zehuan Yuan, and Xiaojuan Qi. Rethinking resolution in the context of efficient video recognition. *Advances in Neural Information Processing Systems*, 35:37865–37877, 2022. 2, 4

[40] Yue Ma, Tianyu Yang, Yin Shan, and Xiu Li. Simvtp: Simple video text pre-training with masked autoencoders. *arXiv preprint arXiv:2212.03490*, 2022. 2

[41] Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. Egoschema: A diagnostic benchmark for very long-form video language understanding. *arXiv preprint arXiv:2308.09126*, 2023. 2, 3, 4, 8

[42] 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. 2, 4

[43] 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. 2

[44] 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

[45] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 3

[46] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477, 2022. 2, 3

[47] Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, 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. 7

[48] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023. 2

[49] 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. 2, 3, 4, 5

[50] 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. 3
[51] Michael S Ryoo, Keerthana Gopalakrishnan, Kumara Kahatapitiya, Ted Xiao, Kanishka Rao, Austin Stone, Yao Lu, Julian Ibarz, and Anurag Arnab. Token turing machines. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19070–19081, 2023. 2
[52] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 4
[53] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 2
[54] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15638–15650, 2022. 2
[55] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022. 2, 3, 4
[56] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2
[57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3
[58] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Luowei Zhou, Yucheng Zhao, Yujie Xie, Ce Liu, Yu-Gang Jiang, and Lu Yuan. Omnivl: One foundation model for image-language and video-language tasks. *Advances in neural information processing systems*, 35:5696–5710, 2022. 2
[59] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2
[60] Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4581–4591, 2019. 2, 4
[61] 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. 2
[62] Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, et al. Language models with image descriptors are strong few-shot video-language learners. *Advances in Neural Information Processing Systems*, 35: 8483–8497, 2022. 2
[63] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894, 2021. 1
[64] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597, 2022. 1, 2
[65] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018. 3
[66] 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. 2, 3
[67] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016. 2, 4
[68] Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5036–5045, 2022. 2
[69] Kashu Yamazaki, Khoa Vo, Quang Sang Truong, Bhiksha Raj, and Ngan Le. Vltint: visual-linguistic transformer-in-transformer for coherent video paragraph captioning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3081–3090, 2023. 8
[70] 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. 2, 4
[71] 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. 1, 2, 3, 4, 7
[72] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *Advances in Neural Information Processing Systems*, 35:124–141, 2022. 1, 2, 3
[73] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*, 2023. 1, 2, 3

[74] 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. 2, 3, 7

[75] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 2

[76] 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. 2

[77] Andy Zeng, Maria Attarian, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S Ryoo, Vikas Sindhwani, Johnny Lee, et al. Socratic models: Composing zero-shot multimodal reasoning with language. In *The Eleventh International Conference on Learning Representations*, 2022. 2, 7

[78] Bowen Zhang, Xiaojie Jin, Weibo Gong, Kai Xu, Zhao Zhang, Peng Wang, Xiaohui Shen, and Jiashi Feng. Multimodal video adapter for parameter efficient video text retrieval. *arXiv preprint arXiv:2301.07868*, 2023. 3

[79] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023. 3

[80] Zhang Zhang and Dacheng Tao. Slow feature analysis for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):436–450, 2012. 3

[81] Luowei Zhou, Chenliang Xu, and Jason Corso. Towards automatic learning of procedures from web instructional videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 2, 3, 4

[82] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 2