Distilling Vision-Language Models on Millions of Videos

Yue Zhao[1,2]* Long Zhao[1] Xingyi Zhou[1] Jialin Wu[1] Chun-Te Chu[1] Hui Miao[1] Florian Schroff[1]
Hartwig Adam[1] Ting Liu[1] Boqing Gong[1] Philipp Krähenbühl[2] Liangzhe Yuan[1]
[1]Google Research [2]University of Texas, Austin

Abstract

*The recent advance in vision-language models is largely attributed to the abundance of image-text data. We aim to replicate this success for video-language models, but there simply is not enough human-curated video-text data available. We thus resort to fine-tuning a video-language model from a strong image-language baseline with synthesized instructional data. The resulting video model by video-instruction-tuning (**VIIT**) is then used to auto-label millions of videos to generate high-quality captions. We show the adapted video-language model performs well on a wide range of video-language benchmarks. For instance, it surpasses the best prior result on open-ended NExT-QA by 2.8%. Besides, our model generates detailed descriptions for previously unseen videos, which provide better textual supervision than existing methods. Experiments show that a video-language dual-encoder model contrastively trained on these auto-generated captions is 3.8% better than the strongest baseline that also leverages vision-language models. Our best model outperforms state-of-the-art methods on MSR-VTT zero-shot text-to-video retrieval by 6%. As a side product, we generate the largest video capation dataset to date.*

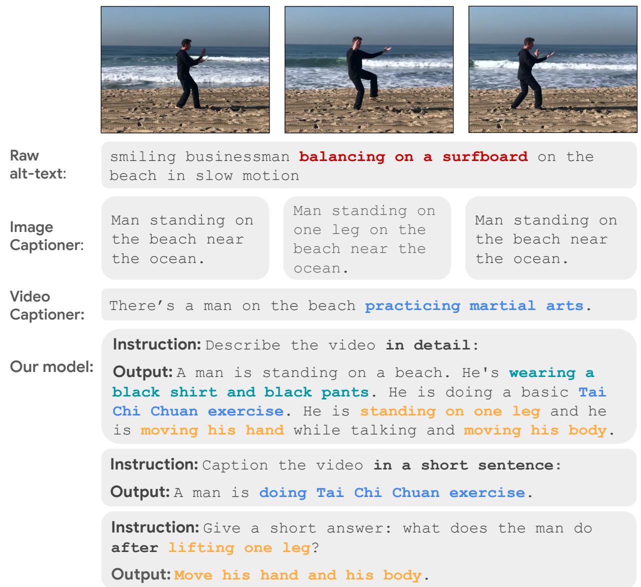

Figure 1. Our video-language model takes a video along with *any* form of instruction as input and generates text according to the instruction. It generates textual descriptions with multiple granularities, including **static appearance**, **general action**, and **detailed body movements**. In contrast, raw alt-text can be **erroneous**; image captioners fail to capture the action; video captioners prefer outputting short text. Our generated data trains a significantly better video-language dual-encoder model. Best viewed in color.

1. Introduction

Much progress in image understanding [15, 45, 59, 75, 81] is fueled by large-scale high-quality image-text datasets [9, 27, 48, 51]. Despite the wide availability on the Internet, annotating videos is nontrivial. For images, humans construct most annotations within 15∼90 seconds per instance [27, 35]. For videos, the annotation time is 1∼2 orders of magnitude higher: it takes about 70 hours to transcribe narratives for a 1-hour video [21, 68] and 700 hours to provide a 1-hour video with instance-level annotations [12]. There have been attempts to automate such a process by retrieving alt-text [2, 42] or transcribing text from audio [40, 76]. However, alt-text can be irrelevant to the video content; audio transcription is often misaligned with the visual information [22]. Recent work [63] leverages existing image-based vision-language models (VLMs). However, the resulting captions are often biased towards static scenes and lose videos' rich temporal information.

In this paper, we propose a simple yet effective approach to adapt an image-based VLM to video and then create high-quality pseudo-captions on millions of videos. As a VLM is generally composed of a visual encoder and a language model, we propose to adapt each component separately to better leverage the relatively small video-text corpora. We first fine-tune the visual encoder on video captioning data while keeping the language component frozen. This adapts the model to dynamic scenes while retain-

*Work done during an internship at Google Research.

ing the diverse ability of the original language decoder. We then fine-tune the language model on a small amount of instruction-following data and keep the visual encoder frozen. This is to emphasize the temporal and causal reasoning ability beyond scene-level description. The resulting video-language model sees both dynamic input and motion-focused output and is capable of generating high-quality pseudo-captions for million-scale web-scraped videos.

Pseudo-captioning by the adapted VLM have the following advantages. First, the captions are generally relevant to visual content because of the maximum likelihood objective during video-captioning training. Second, our pseudo-captions preserve temporal information in videos better than frame-wise captions for videos [38, 63]. Third, the instruction-tuned video-language model generates textual descriptions with multiple granularities, including static appearance, general actions, and detailed body movements. Finally, compared to human labeling, pseudo-captioning is more scalable. For each video, the underlying language model can output multiple candidate captions in parallel in a single pass, and the annotation cost can be further improved given advances in efficient inference techniques [31].

We evaluate the resultant VLM on a wide range of video-language benchmarks, covering video question answering (QA) and captioning, and observe state-of-the-art zero-shot performance on all. For instance, it attains a 29.5% WUPS score on open-ended NExT-QA, 2.8% better than Flamingo-80B while using only $\frac{1}{16}\times$ parameters. We further use this adapted VLM to generate video descriptions on million-scale web-scraped videos. Qualitatively, the generated descriptions are more specific and detailed than alt-text or image captions. To evaluate the pseudo-captions quantitatively, we train a CLIP-style [48] video-language dual-encoder model using the generated descriptions. We observe a striking scaling effect on the performance with respect to the size of pseudo-captioned video data, which does not hold for alt-text alternatives. Our model also works better than the one trained on frame-wise captions followed by LLM summarization. Notably, the dual-encoder model trained on 17 million web-scraped video clips with our machine-generated descriptions achieves the state-of-the-art performance on popular video-text retrieval and video recognition benchmarks. For instance, the model scores 48.4% Recall@1 on MSR-VTT, 6% higher than the best previously reported number.

2. Related Work

Synthetic data from simulators are useful to create new datasets or augment existing ones [13] for vision tasks such as optical flow [14], semantic segmentation [50], and 3D vision [7]. LLM-generated text becomes a great source for language understanding [39]. For example, Vicuna [11] fine-tunes LLaMA [56] on user-shared conversations from ShareGPT. In the context of vision-language understanding, generating high-quality synthetic captions for vision data by leveraging LLMs has been shown effective in improving multimodal datasets for VLMs [43]. VideoChatGPT [38] uses both human-assisted and semiautomatic annotation methods with BLIP-2 [32] and GPT-3.5 to generate high-quality video instruction data. InternVid [63] introduces a scalable approach to automatically construct a high-quality video-text dataset with BLIP-2 and Vicuna. LLaVA [36] incorporates instruction tuning to VLMs, which demonstrates impressive multi-modal chat abilities. However, these methods either focus on image inputs or rely on image models to produce video captions, which fail to capture correct temporal information in videos.

Vision-language models. Utilizing image-text data for pre-training has become the default approach to tackle vision-language tasks. Recently, VLMs based on image-text contrastive learning (*e.g.*, CLIP [48] and ALIGN [26]) attain strong results on zero-shot retrieval and classification tasks. Follow-up studies propose to add more pre-training objectives, such as captioning loss (*e.g.*, CoCa [73]), to enable VLMs to handle different downstream tasks (*e.g.*, image captioning and visual QA). Parallel methods explore leveraging off-the-shelf pre-trained models and keep them frozen during training. They partially freeze either vision or language models (*e.g.*, PaLI [8–10] and LiT [78]) or insert new layers between them (*e.g.*, Flamingo [1] and BLIP-2 [32]) so that the knowledge from frozen models can be transferred to vision and language tasks. Our work builds upon them and tackles video inputs, a more challenging modality involving temporal and causal reasoning of motion.

Video-language models can be adapted from image-language models given that image-based foundation models are pre-trained on web-scale image data. VideoCLIP [67] leverages a pre-trained CLIP model [48] as a frame-level feature extractor and fine-tunes video and text transformers on video datasets. VideoCoCa [69] builds on CoCa [73] and fine-tunes some temporal pooler layers to reason over time. Another line of research focuses on parameter efficient tuning, which is first shown effective on language modeling [30]. AIM [71] adapts pre-trained image models for efficient video understanding by freezing pre-trained weights and tuning a few lightweight adapters. Furthermore, to solve more complex video-language tasks like captioning and QA, researchers leverage the powerful LLMs as a universal interface and adapt LLMs to consume visual tokens. FrozenBiLM [70] leverages a frozen bi-directional language model for video QA. VideoChat [33] and VideoChat-GPT [38] propose a chatbot-like interface to analyze video input. However, VideoChat only shows qualitative analysis while VideoChatGPT relies on a GPT-4 for quantitative evaluation, leading to inconsistency over time. LaViLa [80] develops a video-language model that densely narrates for a video. However, training the narrator assumes videos to be partially annotated. Our work takes a further step and shows that the adapted video-language model generalizes to million-scale *unseen* videos.

3. Preliminaries and Notations

We first describe preliminaries and, meanwhile, introduce some notations facilitating the presentation of our method.

Image-based VLMs take as input an image and a text sequence, which is often called a prompt [4] or an instruction [64], and outputs another text sequence that follows the prompt. Specifically, let $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ denote an input image with height H and width W, $\mathbf{y} = (s_1, \cdots, s_{L_i}) \in \{0,1\}^{L_i \times |\mathbb{S}|}$ the instruction, and $\mathbf{z} = (z_1, \cdots, z_{L_o}) \in \{0,1\}^{L_o \times |\mathbb{S}|}$ the output text that are tokenized [29] into sequences of discrete symbols. Here $\mathbb{S}$ denotes the vocabulary set, and L_i and L_o are the sequence lengths of the instruction and output, respectively.

A typical VLM has a visual encoder F_V and a language model F_L. The visual encoder maps $\mathbf{x}$ to N visual tokens $\mathbf{x}' = F_V(\mathbf{x}) \in \mathbb{R}^{N \times C}$. It is often instantiated by a pre-trained Convolutional Network [23] or Vision Transformer [15] plus an optional projection module in the form of Q-Former [32], Resampler [1], or attentional pooler [73]. The language model projects an input instruction $\mathbf{y}$ to text tokens $\mathbf{y}' \in \mathbb{R}^{L_i \times C}$, concatenates them with the visual tokens, and emits a text sequence recursively $\tilde{z}_l = F_L(\mathbf{x}', \mathbf{y}', \mathbf{z}_{<\ell})$, where $\mathbf{z}_{<\ell} = [\tilde{z}_0, \cdots, \tilde{z}_{l-1}]$ with $\tilde{z}_0$ being a special start-of-sentence token <s>. F_L can be either an encoder-decoder-style model [49, 55], or a decoder-only model [4]. In this paper, we train the VLM using a captioning loss, *i.e.*, the sum of the negative log-likelihood of the correct word at each step:

$$\mathcal{L} = -\sum_{\ell=1}^{L} p(z_\ell | \mathbf{x}', \mathbf{y}', \mathbf{z}_{<\ell}). \tag{1}$$

The key to the recent success of VLMs is the abundance of paired image-text datasets $\{(\mathbf{x}, \mathbf{c})\}$. By setting $\mathbf{y} = \varnothing$ or a fixed task prompt for captioning and $\mathbf{z} = \mathbf{c}$, we can easily scale up VLMs by training on billion-scale datasets [9, 51].

Visual instruction tuning intends to enable VLMs to tackle tasks beyond image captioning [36]. In this case, $(\mathbf{y}, \mathbf{z})$ can be a question-answer pair as in visual QA [20], or more generally, any free-form instruction-answer pair. The paired instruction-answer data are typically transformed from a plain caption via few-shot prompting by a language model [4, 62], *i.e.* $(\mathbf{y}, \mathbf{z}) = \mathrm{LLM}(\mathbf{c})$.

Video-text datasets. One of the main challenges in training video-language models is the lack of video-text data. The largest public video dataset with human-labeled textual descriptions is Spoken Moments in Time (S-MiT) [41], which has ∼500K videos. Although the covered topics are diverse, the video durations are short (2∼3 seconds), and the captions are brief. The textual descriptions are transcribed from audio recordings with inevitable transcription errors. The Video Localized Narratives (VidLN) [57] dataset captures more complex events for longer videos (10∼30 seconds), but it is 10× smaller in the number of videos due to annotation cost. Both lag in scale far behind existing image-text datasets, *e.g*. WebLI-10B and LAION-5B. In the following section, we present an approach to leveraging these existing video-text datasets to efficiently adapt a pre-trained VLM from images to videos so that we can obtain high-quality pseudo-captions for millions of in-the-wild videos. Experiments show our method yields competitive annotation quality and is more scalable than human annotation for videos.

4. Method: Adapting VLMs to Videos

We adapt an image-language model to the video domain in two stages. In the first stage, we adapt the visual encoder while freezing the language component, allowing us to leverage relatively large video-text datasets whose text is unfortunately short and low-quality. In the second stage, we finetune the language encoder and freeze the other model components using a smaller video-text dataset whose text describes the video in detail and provides diversity. We empirically justify the advantage of this two-stage design, which is necessary given the video-text data's quality and size falling behind its image-text counterpart.

4.1. Model

Our video-language model takes a sequence of frames as visual input. Let $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ denote the input video, where T is the number of frames. We pass each frame $\mathbf{x}_t$ into the visual encoder F_V and concatenate all output visual tokens, namely $\mathbf{x}' = [F_V(\mathbf{x}_1), \cdots, F_V(\mathbf{x}_T)] \in \mathbb{R}^{TN \times C}$. By doing so, we maintain the visual modeling capacity from the image-based models [8] and keep both computation and memory cost tractable ($O(TN^2)$ rather than $O(T^2N^2)$). The language model then collects the visual tokens plus input instruction tokens and emits a text sequence.

Model architecture. We start with PaLI-3 [8], a state-of-the-art VLM trained on WebLI [9] which has image-text data only. The visual encoder is a ViT-G/14 [77] with 2B parameters. The language model follows an encoder-decoder architecture based on UL-2 [55] with 3B parameters. We feed the adapted model with 8 frames at 2 FPS and resize the input resolution to 224×224.

4.2. Two-Stage Adaptation

Due to the scarcity of paired video-text data, we propose to fine-tune the video-language model from the image-based baseline in two stages: (1) visual adaptation, where we freeze the language component while fine-tuning the visual part with a relatively large video dataset with short captions; and (2) language adaptation, where we instruction-tune the

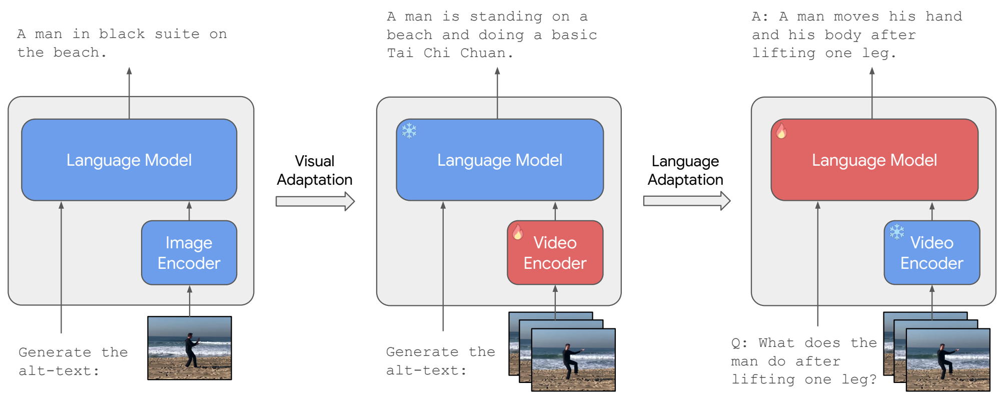

Figure 2. **Overview of adapting vision-language models to videos.** In the first stage of visual adaptation on sequences of video frames, we fine-tune the vision encoder while freezing the language model using a video dataset with captions. In the second stage of language adaptation, we freeze the vision encoder while fine-tuning the language model using a video dataset with instruction-following data, *e.g.* a question that requires temporal reasoning to answer in this example.

language component while freezing the visual part with a smaller video dataset with detailed captions.

Visual adaptation. In the stage of visual adaptation, we fine-tune F_V while keeping F_L frozen using a large video dataset with short captions $\{(\mathbf{x}, \mathbf{c})\}$. We optimize Eq. (1) by setting $\mathbf{y}$ to be a fixed task prompt for captioning ("`Generate the alt-text:`") and $\mathbf{z}$ to be the caption. On one hand, finetuning F_V enables the visual encoder to focus more on scene dynamics rather than appearance. On the other, freezing F_L prevents the language model from possible collapse due to simple text and repetitive patterns.

Language adaptation. In this stage, we fine-tune F_L while keeping F_V frozen using videos with instruction-following data generated as follows. Given a video $\mathbf{x}$ and its caption $\mathbf{c}$, we first prompt an LLM to generate a question $\mathbf{y}$ and the corresponding answer $\mathbf{z}$ which is inferred from the original caption. We optimize Eq. (1) with the $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ triplets.

The video-language model's temporal reasoning ability is highly dependent on the instruction-following data it trains on. To this end, we design prompts to encourage LLMs to generate *causal* and *temporal* questions inspired by how the NExT-QA dataset [65] is constructed. Causal questions either explain the intention of an action that happens first or the cause of an action that occurs next. It typically follows the form of "Why did somebody do something?" or "How did something happen?". Temporal questions ask about the temporal ordering of multiple actions. The temporally ordered actions can either happen on a single object or occur between multiple persons or objects. We provide an example for illustration in Figure 3 and more details in the supplementary materials.

Inference. At inference time, we query the video-language model by feeding sampled video frames for $\mathbf{x}$, the regular task prompt for captioning for $\mathbf{y}$, and a special start-of-sentence token `<s>` for $\mathbf{z} = [z_0]$. We sample from the distribution recursively, i.e. $\tilde{z}_\ell \sim p(z|\mathbf{x}, \mathbf{y}, \tilde{z}_{<\ell})$ until an end-of-sentence token `</s>` is reached. We use nucleus sampling [24], where we only sample from a subset of tokens that contain the vast majority of the probability mass at each step, multiple times. We provide an example of captions before and after video-specific adaptation in Figure 4. Readers can find more results in the supplementary materials in §8. We observe on average 20% longer length in the output sequence after the language adaptation while using the same task prompt for captioning. We attribute it to the effectiveness of instruction tuning.

5. Experiments

First, we summarize the datasets that we use in §5.1. Next, we describe how we harness and evaluate the distilled pseudo-captions in §5.2. We show the main results, *i.e.* (1) the scaling effect of our data generation pipeline, (2) the quality of pseudo-captions by pre-training a dual-encoder model, and (3) the performance of the adapted video-language model on video-language tasks in §5.3. Finally, we discuss the effect of different components in §5.4.

5.1. Datasets

Table 1 summarizes the video datasets used in this paper, and more details are in §9 in the supplementary material. We categorize the datasets into four parts and describe the adaptation data and distilled data first.

Adaptation data. We use two datasets to adapt a vision-language model from images to videos: (1) *Spoken Moments in Times (S-MiT)* [41] contains 500K videos with spoken captions. The videos are typically short (2∼3 seconds) and the transcribed captions are brief (18 words on average per video). It has 481K/8K/3K videos for training/validation/testing. We use the training split to conduct visual adaptation of the video-language model and evaluate the video captioning result by CIDEr score on the

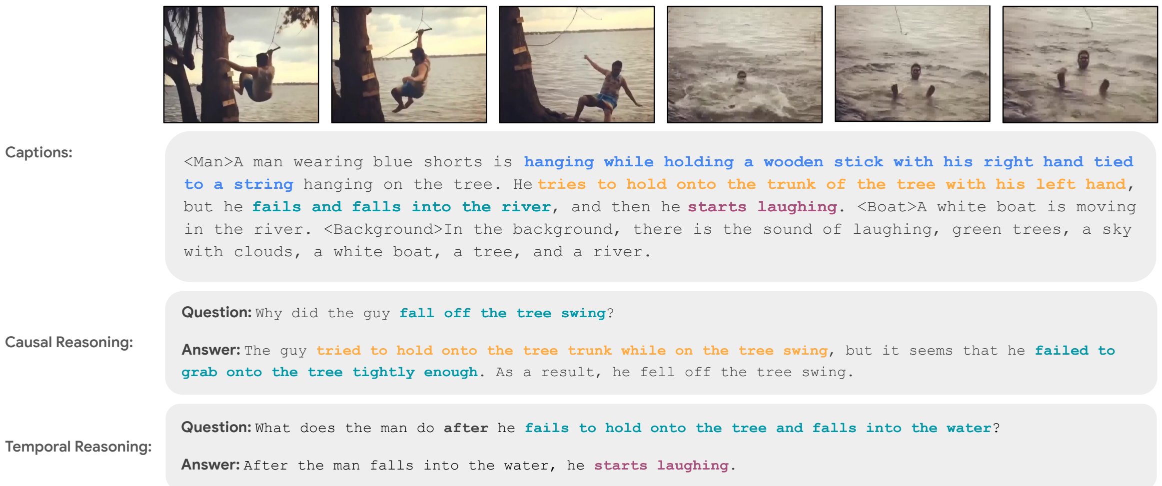

Figure 3. **An example of the instruction-following data.** The first block shows the detailed captions used to prompt an LLM (PaLM 2 [19] in our case), and the following two blocks show the LLM's responses. We show the keyframes in the top block for illustration purpose and do *not* use them while prompting the LLM. Different details in text are highlighted. Best viewed in color.

Figure 4. **An example of video captions by PaLI-3 before and after video-specific adaptation.** We show the keyframes on top for illustration purposes and the generated captions in the following blocks. Different details in text are highlighted. Best viewed in color.

testing split following PaLI [8, 10]. (2) *Video Localized Narratives (VidLN)* [57] annotates comprehensive events in videos which involve multiple actors and possibly actor-actor and actor-object interaction. The narratives are longer (85 words on average) and are better suited to generate a diverse instructing-following corpus. We use the training split which has 47,776 videos from the union of OVIS [46], Oops [17], UVO [58], and Kinetics [5] datasets, to generate instruction-answer pairs for language adaptation.

Data with distilled pseudo-captions. We apply the resultant video-language model to caption two largest-scale webly-scraped video datasets: (1) *VideoCC* [42] contains ∼10M video-caption pairs from 6M unique videos. The raw alt-text is automatically retrieved from those in the Conceptual Captions image-captioning dataset (CC3M) [53] based on image similarity. ∼7.1M clips are available by the time of our experiments. (2) *InternVid* [63] has ∼234M clips from 7M videos. The original captions are synthesized from individual frames' captions by an LLM. We use the publicly available InternVid-10M-FLT subset which has 10M clips with top-scoring video-text similarities. We denote the datasets processed by our method to be **VideoCC**$^{+}$ and **InternVid**$^{+}$. We use both datasets to pre-train a dual-encoder model to show the usefulness of the machine-generated video captions, explained next.

5.2. Harnessing the Distilled Pseudo-Captions

We harness and *evaluate* the distilled pseudo-captions for million-scale web-scraped videos, **VideoCC**$^{+}$ and **InternVid**$^{+}$, using a dual-encoder model [48]. The model's video understanding performance is a solid indicator of the pseudo-captions' quality, and we show that they are of

Dataset	Task	Size	Metrics
S-MiT [41]	ADP	480K (train)	-
VidLN [57]	ADP	47K (train)	-
VideoCC [42]	CP	7M/10M	-
InternVid [63]	CP	10M	-
MSR-VTT [68]	TVR	1K (val, or *1k-A*)	Recall@k
VATEX [60]	TVR	1.5K (test as in [61])	Recall@1
Kinetics-600 [6]	CLS	28K (val)	Accuracy
MSR-VTT [68]	CAP	6.5K(train)+3K(test)	CIDEr
MSR-VTT QA [66]	QA	6.5K(train)+3K(test)	Accuracy
ANet-Captions [28]	CAP	31K(train)+14K(test)	CIDEr
S-MiT [41]	CAP	480K(train)+3K(test)	CIDEr
ANet-QA [74]	QA	32K(train)+8K(test)	Accuracy
NExT-OE-QA [65]	QA	37K(train)+9K(test)	Wu-Palmer Similarity (WUPS)

Table 1. **Dataset summary.** ADP is short for adapting VLMs while CP is for contrastive pre-training a dual-encoder model. Evaluation tasks include text-to-video retrieval (TVR), action classification (CLS), video captioning (CAP), and video question answering (QA).

higher quality than the original text in VideoCC and InternVid.

Contrastive training of a dual-encoder model. We train a video-language dual-encoder model like CLIP [48]. Specifically, given the input video frames $\mathbf{x}$ and machine-generated captions $\tilde{\mathbf{c}}$, the model applies a visual encoder G_V plus a projection head h_V and a text encoder G_T plus a projection head h_T in parallel to obtain the global visual and textual embedding, respectively,

$$\mathbf{u} = h_V(G_V(\mathbf{x})), \mathbf{v} = h_T(G_T(\tilde{\mathbf{c}})). \quad (2)$$

We use the InfoNCE [44] loss to train the model. Note that we deliberately choose a different notation $G_{(\cdot)}$ than $F_{(\cdot)}$ in the VLM in §3 because the dual-encoder model does *not* share any parameters with the VLM.

Model architecture. The dual-encoder model has a vision encoder and a text encoder. The video input is represented by 4 frames at 2 FPS. The vision encoder is a Vision Transformer [15] with joint spatial-temporal attention (denoted as "ViT-*st*") following [79]. We use ViT-L/14 to report the main result and ViT-B/16 for ablation studies if not otherwise specified. The weights are initialized from CLIP [48] except that we randomly initialize the temporal position embedding $\mathrm{PE}_t \in \mathbb{R}^{T\times D}$ and add it to the original spatial position embedding $\mathrm{PE}_s \in \mathbb{R}^{N\times D}$, *i.e.* $\mathrm{PE}[i,:,:] = \mathrm{PE}_t[i,\texttt{None},:] + \mathrm{PE}_s[\texttt{None},:,:]$. The text encoder is a 12-layer GPT-like Transformer [47]. It takes as input one video caption, tokenizes it using BPE [52], and keeps at most 77 tokens. If a video has more than one caption, we randomly sample one of them at each time.

5.3. Main Results

We report the dual-encoder model's text-to-video retrieval performance (on MSR-VTT and VATEX) and video classification accuracy (on Kinetics-600), both under the *zero-shot* setting. These results are meant to evaluate the quality of the distilled video pseudo-caption data. Besides, we also evaluate the VLM adapted to the video domain on a few representative video-language benchmarks following PaLI-3 [8], including video captioning (MSR-VTT [68], ActivityNet-Captions [28]) and video question-answering (MSR-VTT QA [66], ActivityNet QA [74], and NExT Open-Ended QA [65]). We enumerate the datasets involved at the bottom of Table 1 and leave details in §9.

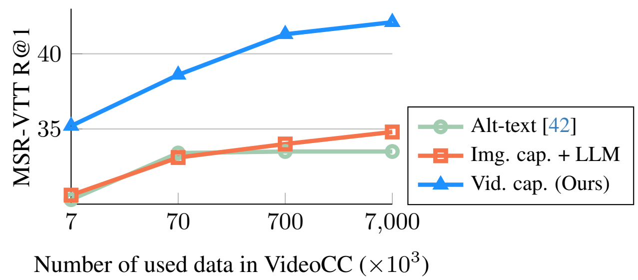

Figure 5. **Scaling effect of video captioning.** For VLM-generated captions, the zero-shot video retrieval performance consistently improves with respect to an increasing amount of video data. Pre-training on retrieved alt-text quickly stagnates.

Distilled vs. alt-text captions at various scales. Figure 5 shows that the distilled pseudo-captions for VideoCC outperform VideoCC's original Alt-text captions, by a striking margin, when the dual-encoder models trained using different subsets of VideoCC are evaluated on the MSR-VTT retrieval task. We find that Recall@1 quickly saturates when training the dual-encoder model on VideoCC with alt-text. Specifically, training with only 1% VideoCC$^+$ (∼70K) achieves the same level of Recall@1 with training with the whole VideoCC set (∼7M), indicating that the original alt-text scales poorly. We attribute the alt-text's inferior performance to a compounding error of textual noise [26], spurious correlation when computing visual similarities [72], and the visual discrepancy between images and videos. In contrast, training the dual-encoder model with the pseudo-captions clearly exhibits a pleasant scaling effect: R@1 consistently increases with more pre-training video data. We also include in Figure 5 the curve corresponding to the pseudo-captions distilled from the image-language model before it is adapted to the video domain. It almost overlaps with the alt-text curve at the beginning and then becomes slightly better near the end.

Distilled captions for video understanding. We continue to evaluate the distilled pseudo-captions by the corresponding dual-encoder model's *zero-shot* performance on text-to-video retrieval and video classification. From Table 2, we see that the pseudo-captions distilled from our VLM significantly improve the dual-encoder over the original text in VideoCC and InternVid. On VideoCC, with all other settings being the same, the dual-encoder model trained on VideoCC$^+$, achieves 48.2% Recall@1 on MSR-VTT, 11.2% better than the one trained on the original Alt-text. It also clearly surpasses the recent ViCLIP trained

Method	Pre-training Dataset	MSR-VTT TVR			VATEX TVR			Kinetics-600	
		R@1	R@5	R@10	R@1	R@5	R@10	Top-1	Top-5
CLIP [48]	WIT	31.2	53.7	64.2	45.2	-	-	55.1	79.2
CLIP4Clip [37]	WIT	30.6	54.4	64.3	-	-	-	-	-
CLIP4Clip [37]	WIT→VideoCC (10M)	33.7	57.9	67.9	-	-	-	-	-
InternVideo [61]	WIT→Mixed (12M)	40.0	65.3	74.1	49.5	79.7	87.0	-	
ViCLIP [63]	WIT→WebVid (10M)	35.6	-	-	-	-	-	58.7	81.0
ViCLIP [63]	WIT→InternVid (10M)	42.4	-	-	-	-	-	62.2	84.9
	WIT→VideoCC	37.0	62.1	72.5	37.7	66.9	77.2	48.6	74.8
	WIT→VideoCC$^+$ (**Ours**)	48.2	72.2	80.8	64.2	90.2	95.1	61.1	85.6
	Absolute gain Δ	**+11.2**	**+10.1**	**+8.3**	**+26.5**	**+23.3**	**+17.9**	**+12.5**	**+10.8**
CLIP (ViT-*st*-L)	WIT→InternVid	42.5	67.0	76.8	58.7	87.0	93.0	60.7	85.0
	WIT→InternVid$^+$ (**Ours**)	46.3	71.5	80.3	65.2	91.3	95.5	62.7	86.2
	Absolute gain Δ	**+3.8**	**+4.5**	**+3.5**	**+6.5**	**+4.3**	**+2.5**	**+2.0**	**+1.2**
	WIT→VideoCC$^+$+InternVid$^+$ (**Ours**)	**48.4**	**73.5**	**81.9**	**65.6**	**91.7**	**95.8**	**62.8**	**86.4**

Table 2. **Zero-shot text-to-video retrieval performance on MSR-VTT & VATEX and video recognition performance on Kinetics-600 using different sources of textual descriptions.** $\mathcal{D}^+$ means that the captions in the video dataset $\mathcal{D}$ are generated by our proposed pipeline. $\mathcal{D} \in \{\text{VideoCC}, \text{InternVid}\}$ in our experiments.

Method	Pre-training Dataset	MSR-VTT		ActivityNet		NExT-OE-QA
		Caption	QA (Acc.)	Caption	QA (Acc.)	QA (WUPS)
Prior SOTA	-	18.6 DeCap [34]	16.8 FrozenBiLM [70]	15.0 DeCap [34]	25.9 FrozenBiLM [70]	26.7 Flamingo [1]
PaLI-3$_{8f}$ [9]	WebLI	21.3	12.7	13.8	22.9	23.2
Ours	WebLI→SMiT+VidLN	**48.2**	**24.4**	**31.0**	**29.6**	**29.5**

Table 3. **Zero-shot performance of the Video-Language Model on video-language understanding tasks.** Our adapted video-language model significantly improves over the 8-frame PaLI-3 baseline and outperforms the best reported numbers.

on InternVid, which contains 2× more unique videos than VideoCC. On InternVid, our model trained on InternVid$^+$ is 3.8% better than the baseline trained on the original InternVid's auto-generated captions. It is worth noting that our adapted VLM is also "lighter-weight" compared to the multi-scale captioning pipeline in InternVid [63], which relies on both image captioning models (BLIP-2) [32] on multiple frames and an LLM to put them together. We also highlight the zero-shot top-1 and top-5 classification accuracy on Kinetics-600. For instance, the dual-encoder model trained on VideoCC$^+$/InternVid$^+$ improves the baselines on VideoCC/InternVid by 12.5/2.0% top-1 accuracy.

Interestingly, we notice that the model trained on InternVid$^+$ works better on action recognition, while the one trained on VideoCC$^+$ is better on video retrieval. This is probably because the InternVid videos are specifically collected based on action phrases [63], while VideoCC is seeded from image-captioning data [42]. Since the two datasets are complementary, combining them indeed leads to performance gains as shown in the last row in Table 2.

Evaluating the video-language model. We compare the adapted VLM with the baseline PaLI-3 in Table 3. We focus on the zero-shot performance where we apply the model to the testing split of downstream tasks *without* any tuning. This setting resembles the scenario where we generate pseudo-captions on VideoCC and InternVid, and it provides us with a direct measure on well-established benchmarks. Specifically, the greatly improved CIDEr score on MSR-VTT and ActivityNet-Captions showcases the effectiveness of adapting a VLM to the video domain. We also see excellent zero-shot question-answering results compared to PaLI-3. On the challenging open-ended NExT-QA dataset, our model outperforms Flamingo [1] by 2.8% (WUPS score). This gain is achieved using only $\frac{1}{16}\times$ of the parameters (5B *vs* 80B) and $\frac{1}{50}\times$ of training videos (0.55M publicly available S-MiT&VidLN *vs* 27M in-house VTP). On MSR-VTT QA and ActivityNet QA, our adapted model achieves 7.6% and 3.7% higher accuracy than FrozenBiLM [70], trained on WebVid-10M [2].

5.4. Ablation Studies

What makes captioning better? We investigate the key to generating better captions for contrastive pre-training video-language dual-encoder models in Table 4. The comparison starts from the alt-text-only baseline which achieves 37.0% text-to-video R@1 on MSR-VTT. Using frame-level captions produced by PaLI-3 *as-is* increases R@1 by 2.5%. We also attempt to merge multiple frames' captions into a single sentence with PaLM-2 [19] similar to the pipeline in InternVid [63] but see marginal gain (0.3%). This result is consistent with our observation that LLMs often fail to interpolate when key temporal information is lost in the

PaLI-3	LLM	Adapting VLM (§4.2) Visual	Language	Multi. Samples	MSR-VTT Recall@1
					37.0
✓					39.5 (+2.5)
✓	✓				39.8 (+2.8)
✓		✓			41.7 (+4.7)
✓		✓		✓	43.6 (+6.6)
✓		✓	✓	✓	44.3 (+7.3)

Table 4. **The effect of using different sources of textual descriptions.** The captioning quality is measured by the zero-shot text-to-video retrieval performance (Recall@1) on MSR-VTT. The first line with no components checked refers to the alt-text baseline. The "LLM"-column means that we use PaLM 2 [19] to summarize captions from multiple frames similar to [63].

Visual Adaptation F_V	Self-training	F_L	S-MiT Caption (CIDEr)
✗		✓	41.2
✓		✗	42.3
✓		✓	40.3
✓	✓	✗	43.5

Table 5. **Adapting vision encoder.** ✓ and ✗ denote fine-tuning and freezing the parameters respectively. Fine-tuning the visual part while freezing the language model yields better results.

image-level descriptions. We also encounter a trade-off between being concise but lacking diversity and being detailed but vulnerable to hallucination. If we conduct visual adaptation in PaLI-3, the resulting video captions almost double the gain from 2.5% to 4.7%. Generating multiple captions independently with nucleus sampling contributes 1.9%. Finally, doing language adaptation on PaLI-3 with instruction-following data further improves R@1 by 0.7%.

How should we do visual adaptation? We study several ways for visual adaptation in Table 5. The first option, *i.e.* freezing the visual encoder F_V while fine-tuning the language model F_L, takes inspiration from LiT [78]. This leads to a drop of 1.1 CIDEr score compared to our default recipe, where F_V is fine-tuned and F_L frozen. We ascribe it to the visual discrepancy between images and videos: The downstream tasks in [9, 78] are mostly still images, the same as the large-scale pre-training data. In contrast, the videos of our interests have unique characteristics such as object movement, camera motion, and the resultant visual degradation. We also observe a performance drop if we fine-tune both F_V and F_L. This recipe may be prone to over-fitting because the video-text dataset lacks diversity and quantity. Finally, we show that self-training with VideoCC pseudo-captions (details in §11.3) improves captioning results by 1.2 CIDEr score, reaching 43.5. It is worth noting that this number is on par with the best-performing PaLI-X [10] which has 11× more parameters and takes 2× more frames as input than ours.

How should we do language adaptation? We study the effect of instruction-following data in Table 6 when do-

Instruction data	MSR-VTT Caption (CIDEr)	ActivityNet Caption (CIDEr)	NExT-OE QA (WUSP)
None (PaLI-3)	21.3	13.8	23.2
LLaVA 1.0 [36]	16.9	25.1	16.3
ActivityNet-Instruct [38]	30.8	34.6	11.7
Ours			
+ VidLN Causal/temporal Reasoning	28.5	29.5	5.0
+ SMiT Captions	**51.6**	**35.1**	3.9
+ VidLN Short-QA	48.2	31.0	**29.5**

Table 6. **Effect of instruction data.** Our proposed instruction data benefits the adaptation of the video-language model, reflected by better zero-shot captioning results and QA accuracy.

ing language adaptation. We start with some representative visual instructional tuning datasets. The first is LLaVA-1.0 [36] with 150K instruction pairs. We find that it improves the CIDEr score by 7.3 on ActivityNet Captions but decreases by 4.4 on MSR-VTT Captions. The second is ActivityNet-Instruct [38] with 100K instruction pairs from ActivityNet-Captions [28]. It improves CIDEr score on both MSR-VTT and ActivityNet Captions, indicating that video-specific instructional-following data is essential to video-language tasks. We then conduct an incremental study on our LLM-prompted instructional corpus on VidLN+SMiT by adding one component at a time. First, we fine-tune the language model with only reasoning data. The adapted model works on par with the one fine-tuned on ActivityNet-Instruct on ActivityNet-Captions even without seeing ActivityNet videos, demonstrating the generalization of our instructed data. Next, we include the captioning data on S-MiT and see a higher CIDEr score on MSR-VTT and ActivityNet Caption. However, both models suffer from significant degradation in zero-shot QA accuracy. This is expected since the answers in all existing video QA datasets are typically short (1∼3 words) while our instructional data typically contains detailed reasoning (Figure 3). To mitigate the gap, we further add QA pairs that are few-shot prompted based on Oops-QA [57], and prepend the question with a QA-specific task prompt ("`Answer in en:`"). The final model restores its zero-shot question-answering ability at the cost of a slight performance drop in captioning.

More ablations. We leave more ablations and discussions in the supplementary materials.

6. Conclusion

We present an approach to adapting an image-based vision-language model to videos and distilling high-quality pseudo-captions for millions of videos. The adapted video-language model obtains excellent zero-shot performance on various video-language benchmarks. The pseudo-captions yield a stronger dual-encoder model and show positive scaling behavior with respect to the number of videos.

Acknowledgements. This material is based upon work in part supported by the National Science Foundation under Grant No. IIS-1845485.

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. In *NeurIPS*, 2022. 2, 3, 7

[2] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 1, 7

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019. 3

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 3

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 5

[6] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 6, 1

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[8] Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, Daniel Salz, Xi Xiong, Daniel Vlasic, Filip Pavetic, Keran Rong, Tianli Yu, Daniel Keysers, Xiaohua Zhai, and Radu Soricut. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023. 2, 3, 5, 6, 1

[9] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. In *ICLR*, 2023. 1, 3, 7, 8

[10] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. On scaling up a multilingual vision and language model. In *CVPR*, 2024. 2, 5, 8, 1

[11] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 2

[12] Ahmad Darkhalil, Dandan Shan, Bin Zhu, Jian Ma, Amlan Kar, Richard Higgins, Sanja Fidler, David Fouhey, and Dima Damen. Epic-kitchens visor benchmark: Video segmentations and object relations. In *NeurIPS D&B*, 2022. 1

[13] Celso M de Melo, Antonio Torralba, Leonidas Guibas, James DiCarlo, Rama Chellappa, and Jessica Hodgins. Next-generation deep learning based on simulators and synthetic data. *Trends in cognitive sciences*, 2022. 2

[14] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 2

[15] 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 *ICLR*, 2021. 1, 3, 6

[16] Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-sourced webly-supervised learning for video recognition. In *ECCV*, 2020. 3

[17] Dave Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video. In *CVPR*, 2020. 5

[18] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019. 3

[19] Google. Palm 2 technical report, 2023. 5, 7, 8

[20] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 3

[21] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022. 1

[22] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *CVPR*, 2022. 1

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[24] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *ICLR*, 2020. 4

[25] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python. 2020. 1

[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 *ICML*, 2021. 2, 6

[27] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, Dollár Piotr, and Girshick Ross. Segment anything. In *ICCV*, 2023. 1

[28] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 6, 8, 1

[29] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*, 2018. 3

[30] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *EMNLP*, 2021. 2

[31] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *ICML*, 2023. 2

[32] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 2, 3, 7

[33] 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

[34] Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. Decap: Decoding clip latents for zero-shot captioning via text-only training. In *ICLR*, 2023. 7

[35] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2, 3, 8

[37] 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*, 2022. 7

[38] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*, 2023. 2, 8

[39] Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. Generating training data with language models: Towards zero-shot language understanding. In *NeurIPS*, 2022. 2

[40] 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 *ICCV*, 2019. 1

[41] Mathew Monfort, SouYoung Jin, Alexander Liu, David Harwath, Rogerio Feris, James Glass, and Aude Oliva. Spoken moments: Learning joint audio-visual representations from video descriptions. In *CVPR*, 2021. 3, 4, 6

[42] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manen, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions. In *ECCV*, 2022. 1, 5, 6, 7

[43] Thao Nguyen, Samir Yitzhak Gadre, Gabriel Ilharco, Sewoong Oh, and Ludwig Schmidt. Improving multimodal datasets with image captioning. In *NeurIPS D&B*, 2023. 2

[44] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 6

[45] OpenAI. Gpt-4v(ision) system card, 2023. 1

[46] Jiyang Qi, Yan Gao, Yao Hu, Xinggang Wang, Xiaoyu Liu, Xiang Bai, Serge Belongie, Alan Yuille, Philip HS Torr, and Song Bai. Occluded video instance segmentation: A benchmark. *IJCV*, 2022. 5

[47] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019. 6

[48] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 5, 6, 7

[49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 2020. 3

[50] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 2

[51] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS D&B*, 2022. 1, 3

[52] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *ACL*, 2016. 6

[53] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 5

[54] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 3

[55] Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. Ul2: Unifying language learning paradigms. In *ICLR*, 2023. 3

[56] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[57] Paul Voigtlaender, Soravit Changpinyo, Jordi Pont-Tuset, Radu Soricut, and Vittorio Ferrari. Connecting vision and language with video localized narratives. In *CVPR*, 2023. 3, 5, 6, 8

[58] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021. 5

[59] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *CVPR*, 2023. 1

[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 *ICCV*, 2019. 6, 1

[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. 6, 7, 1, 2

[62] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. In *ACL*, 2023. 3

[63] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. In *ICLR*, 2024. 1, 2, 5, 6, 7, 8

[64] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *ICLR*, 2022. 3

[65] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. Next-qa: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021. 4, 6, 1

[66] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM MM*, 2017. 6, 1

[67] 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 *EMNLP*, 2021. 2

[68] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 1, 6

[69] 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, 1

[70] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *NeurIPS*, 2022. 2, 7

[71] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. Aim: Adapting image models for efficient video action recognition. In *ICLR*, 2023. 2

[72] Yu Yang, Besmira Nushi, Hamid Palangi, and Baharan Mirzasoleiman. Mitigating spurious correlations in multimodal models during fine-tuning. In *ICML*, 2023. 6

[73] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *TMLR*, 2022. 2, 3

[74] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, 2019. 6, 1

[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. 1

[76] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *CVPR*, 2022. 1

[77] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022. 3

[78] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. LiT: Zero-shot transfer with locked-image text tuning. In *CVPR*, 2022. 2, 8

[79] Yue Zhao and Philipp Krähenbühl. Training a large video model on a single machine in a day. *arXiv preprint arXiv:2309.16669*, 2023. 6

[80] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *CVPR*, 2023. 2

[81] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, 2022. 1