Learning Video Representations from Large Language Models

Yue Zhao[1,2*] Ishan Misra[1] Philipp Krähenbühl[2] Rohit Girdhar[1]
[1]FAIR, Meta AI [2]University of Texas, Austin
facebookresearch.github.io/LaViLa

Abstract

We introduce **LaViLa***, a new approach to learning video-language representations by leveraging Large Language Models (LLMs). We repurpose pre-trained LLMs to be conditioned on visual input, and finetune them to create automatic video narrators. Our auto-generated narrations offer a number of advantages, including dense coverage of long videos, better temporal synchronization of the visual information and text, and much higher diversity of text. The video-text embedding learned contrastively with these additional auto-generated narrations outperforms the previous state-of-the-art on multiple first-person and third-person video tasks, both in zero-shot and finetuned setups. Most notably,* LaViLa *obtains an absolute gain of* **10.1%** *on EGTEA classification and* **5.9%** *Epic-Kitchens-100 multi-instance retrieval benchmarks. Furthermore,* LaViLa *trained with only half the narrations from the Ego4D dataset outperforms baseline models trained on the full set, and shows positive scaling behavior on increasing pre-training data and model size.*

1. Introduction

Learning visual representation using web-scale image-text data is a powerful tool for computer vision. Vision-language approaches [31, 49, 80] have pushed the state-of-the-art across a variety of tasks, including zero-shot classification [49], novel object detection [87], and even image generation [52]. Similar approaches for videos [4, 39, 46], however, have been limited by the small size of paired video-text corpora compared to the billion-scale image-text datasets [31, 49, 84]—even though access to raw video data has exploded in the past decade. In this work, we show it is possible to *automatically* generate text pairing for such videos by leveraging Large Language Models (LLMs), thus taking full advantage of the massive video data. Learning video-language models with these automatically generated annotations leads to stronger representations, and as Figure 1 shows, sets a new state-of-the-art on six popular first and third-person video benchmarks.

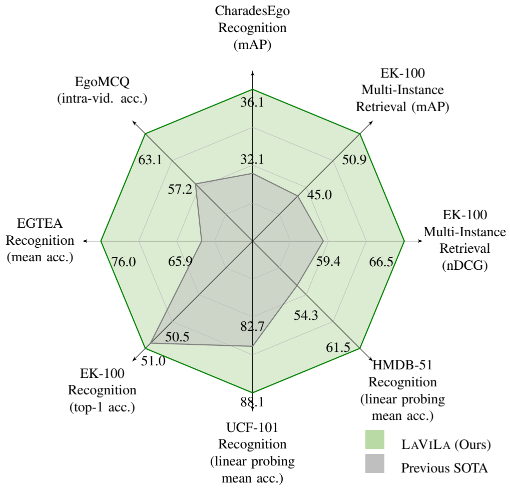


Figure 1. **LaViLa sets a new state-of-the-art** across a number of first and third-person video understanding tasks (*cf*. Table 1 for details), by learning a video-language representation using supervision from large language models as narrators.

Our method, called **LaViLa**: **La**nguage-model augmented **Vi**deo-**La**nguage pre-training, leverages pre-trained LLMs, *e.g*. GPT-2 [50], which encode within their weights a treasure trove of factual knowledge and conversational ability. As shown in Figure 2, we repurpose these LLMs to be "visually-conditioned narrators", and finetune on all accessible paired video-text clips. Once trained, we use the model to densely annotate thousands of hours of videos by generating rich textual descriptions. This pseudo-supervision can thus pervade the entire video, in between and beyond the annotated snippets. Paired with another LLM trained to rephrase existing narrations, LaViLa is able to create a much larger and more diverse set of text targets for video-text contrastive learning. In addition to setting a new state-of-the-art as noted earlier, the stronger representation learned by LaViLa even outperforms prior work using only half the groundtruth annotations (Figure 5).

LaViLa's strong performance can be attributed to a number of factors. First, LaViLa can provide temporally

[*] Work done during an internship at Meta.

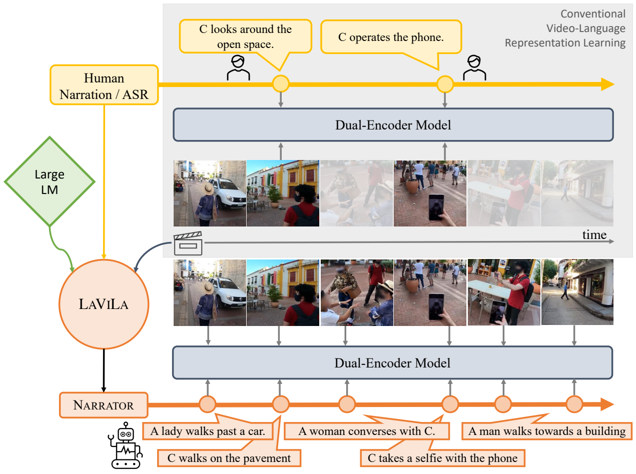


Figure 2. **LaViLa** leverages Large Language Models (LLMs) to densely narrate long videos, and uses those narrations to train strong dual-encoder models. While prior work uses sparsely labeled text by humans, or weakly aligned text transcribed from speech, LaViLa is able to leverage dense, diverse, and well-aligned text generated by a LLM.

dense supervision for long-form videos, where the associated captions are either too sparse, or the video-level "Alt-Text" (in the case of web videos) does not describe all the nuanced activities happening in it. Second, the generated text is well-aligned with the visual input. Although prior work has leveraged automatic speech transcription on HowTo videos [45] to automatically extract clips paired with text from the speech, such datasets have relatively poor alignment between the visual and textual content ($\leq 50\%$, *cf*. [25, 45]), limiting the quality of the learned representations. Third, LaViLa can significantly expand annotations when only a little is available. For instance, videos of mundane day-to-day activities, especially from an egocentric viewpoint, could be very useful for assistive and augmented reality applications. Such videos, however, are rare on the internet, and hence do not readily exist with associated web text. Recent work [24] instead opted to manually capture and narrate such video data. These narrations however required significant manual effort: 250K hours of annotator time spent in narrating 3.6K hours of video. In contrast, LaViLa is able to automatically narrate each video multiple times and far more densely, and hence learns much stronger representations.

We extensively evaluate LaViLa across multiple video-text pre-training datasets and downstream tasks to validate its effectiveness. Specifically, after being pre-trained on Ego4D, the largest egocentric video datasets with narrations, LaViLa can re-narrate the whole dataset 10× over. The resulting model learned on these expanded narrations sets a new state-of-the-art on a wide range of downstream tasks across challenging datasets, including multi-instance video retrieval on Epic-Kitchens-100 (**5.9%** absolute gain on mAP), multiple-choice question answering on Ego4D (**5.9%** absolute gain on intra-video accuracy), and action recognition on EGTEA (**10.1%** absolute gain on mean accuracy). It obtains gains both when evaluated for zero-shot transfer to the new dataset, as well as after fine-tuning on that dataset. Similar gains are shown in third-person video data. When training LaViLa after densely re-narrating HowTo100M, we outperform prior work on downstream action classification on UCF-101 and HMDB-51. In a case study of semi-supervised learning, we show that our model, which only ever sees 50% of the human-labeled data, is capable of outperforming the baseline model trained with all the narrations. Moreover, the gains progressively increase as we go to larger data regimes and larger backbones, suggesting the scalability of our method.

2. Related Work

Vision-language representation learning maps visual and textual embeddings into a common space using metric-learning techniques [21, 73]. Recently, different pretext tasks are proposed to learn a finer-grained association between visual and textual modality, *e.g*. masked language modeling (MLM) [10, 41, 62] and captioning [16, 80]. Another line of research focuses on scaling up both models and pre-training data. For instance, CLIP [49] is pre-trained on 400M image-text pairs with a contrastive loss (InfoNCE [48, 59]) while CoCa [80] unifies contrastive and generative approaches with a single foundation model. Similar trends are also witnessed in the video-text domain [36, 64, 88]. However, collecting high-quality video-text data is more difficult than image-text. Therefore, efforts are made to learn from uncurated videos with machine-generated audio transcripts via contrastive learning [44, 77, 82] or unsupervised alignment [25] while other works focus on either adapting well-performing image-text models to videos [32, 40, 47, 78], or curriculum learning from a single frame to multiple frames [4]. In contrast, our approach leverages language models to generate temporally dense textual supervision on long-form videos.

Generative Visual Language Models (VLM) were first used for image/video captioning using recurrent networks [17, 68] and Transformer-based architectures [42, 56]. More recently, generative VLMs have unified multiple vision tasks [11, 89] by training multi-modal Transformers on visual-text pairs [30, 81]. Meanwhile, generative VLMs also excel at multimodal tasks via zero-shot or few-shot prompting [1, 65, 83] by leveraging multi-billion-parameter LLMs pre-trained on massive text corpus [7, 28, 50]. In our work, we demonstrate that generative VLMs can narrate long videos and the resulting video-text data benefits video-language representation learning.

Large-scale multimodal video datasets are crucial for video understanding tasks but are hard to collect. Conventional video-text datasets [8, 53, 86] either have limited scenarios, *e.g*. cooking, or are not large enough to learn generic video representation. Miech *et al*. [45] scrape over 100 million video-text pairs via automatic audio transcription from long-form How-To videos. However, ASR introduces noticeable textual noise and visual-text unalignment [25]. WebVid [4] contains 10 million short videos with textual descriptions. But it is still several orders of magnitude smaller than the image counterparts [49, 54] and is harder to scale up since it is sourced from stock footage sites. The recently released Ego4D [24] dataset offers 3,600 hours of egocentric videos in which written sentence narrations are manually annotated every few seconds but requires significant manual effort. In contrast, our method shows a promising alternative by automatically narrating videos using supervision from LLM.

Data augmentation techniques in NLP, including word-level replacement based on synonyms [72, 85] or nearest-neighbor retrieval [19, 70], improve text classification accuracy. We refer readers to [20] for a comprehensive survey. In this paper, we show that sentence-level paraphrasing based on text-to-text models [51] is helpful for video-language pre-training.

3. Preliminaries

A video V is a stream of moving images I. The number of frames $|V|$ can be arbitrarily long while video models typically operate on shorter clips, which are often in the range of a few seconds. Therefore, we skim through a long-form video and represent it by a set of N short clips, *i.e*. $\mathcal{X}$. Each clip x_i is defined by a specific start and end frame $x_i = \{I_{t_i}, \cdots, I_{e_i}\}$, where $0 < t_i < e_i \leq |V|$, and is typically associated with some annotation y_i. This annotation could be a class label or free-form textual description of the clip. We denote a video by the set of annotated clips with their corresponding annotations, *i.e*. $(\mathcal{X}, \mathcal{Y}) = \{(x_1, y_1), \cdots, (x_N, y_N)\}$. Note that the annotated clips often cannot densely cover the entire video due to the annotation cost and visual redundancy, *i.e*. $\bigcup_i [t_i, e_i] \subsetneq [0, |V|]$.

A typical video model $\mathcal{F}(\mathcal{X}, \mathcal{Y})$ learns from these clip-level annotations using a standard training objective such as a cross-entropy loss when the annotations are class labels with a fixed vocabulary. However, more recently, dual-encoder-based contrastive approaches like CLIP [49, 77] have gained popularity. They work with free-form textual annotations which are tokenized [55] into sequences of discrete symbols, *i.e*. $y = (s_1, s_2, \cdots, s_L) \in \{1, 0\}^{|\mathbb{S}| \times L}$. The model consists of a visual encoder $f_\mathrm{v} : \mathbb{R}^{T \times 3 \times H \times W} \mapsto \mathbb{R}^{D_\mathrm{v}}$ plus a projection head $h_\mathrm{v} : \mathbb{R}^{D_\mathrm{v}} \mapsto \mathbb{R}^d$ and a textual encoder $f_\mathrm{t} : \{1, 0\}^{|\mathbb{S}| \times L} \mapsto \mathbb{R}^{D_\mathrm{t}}$ plus a projection head $h_\mathrm{t} : \mathbb{R}^{D_\mathrm{t}} \mapsto \mathbb{R}^d$ in parallel to obtain the global visual and textual embedding respectively:

$$\mathbf{v} = h_\mathrm{v}(f_\mathrm{v}(x)), \quad \mathbf{u} = h_\mathrm{t}(f_\mathrm{t}(y)).$$

A contrastive loss, such as InfoNCE [48], learns global embeddings that associate corresponding video and text embeddings within a batch of samples $\mathcal{B}$,

$$\frac{1}{|\mathcal{B}|} \sum_{(x,y) \in \mathcal{B}} \left(\text{InfoNCE}(\mathbf{v}, \mathbf{u}) + \text{InfoNCE}(\mathbf{u}, \mathbf{v}) \right). \tag{1}$$

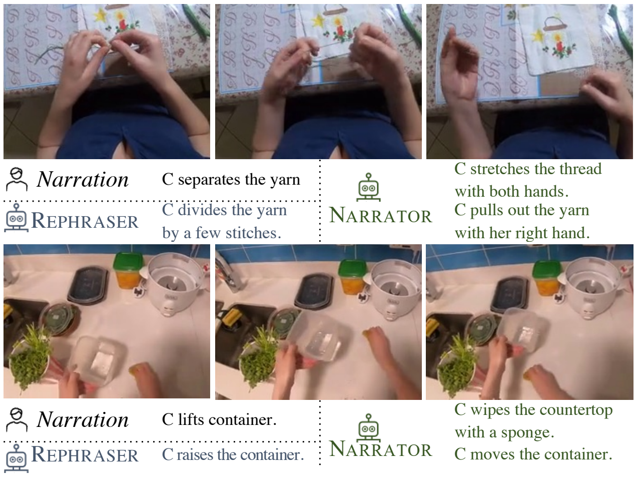

Narration
C separates the yarn
REPHRASER
C divides the yarn by a few stitches.
NARRATOR
C stretches the thread with both hands.
C pulls out the yarn with her right hand.
Narration
C lifts container.
REPHRASER
C raises the container.
NARRATOR
C wipes the countertop with a sponge.
C moves the container.

Figure 3. **Generated samples by our NARRATOR and REPHRASER.** NARRATOR generates new descriptions of the action taking place, potentially focusing on other objects being interacted with. REPHRASER not only changes the word order of the human narration but also diversifies it by using related verbs or nouns.

4. LAVILA

In LAVILA, we leverage large language models (LLMs) as supervision to train the dual-encoder model, where the LLMs serve as vision-conditioned narrators and automatically generate textual descriptions from video clips. In particular, we exploit supervision from two LLMs: (1) **NARRATOR** (§ 4.1) is a *visually-conditioned* LLM that pseudo-labels existing and new clips with narrations, generating new annotations $(\mathcal{X}', \mathcal{Y}')$. (2) **REPHRASER** (§ 4.2) is a standard LLM that paraphrases narrations in existing clips, augmenting those annotations to $(\mathcal{X}, \mathcal{Y}'')$. As illustrated in Figure 3, NARRATOR generates new descriptions of the action taking place, potentially focusing on other objects being interacted with. REPHRASER serves to augment the text input, *e.g*., changes word order of the human narration and additionally replaces common verbs or nouns, making annotations more diverse. Finally, we train the **dual-encoders** (§ 4.3) on all these annotations combined, *i.e*. $(\mathcal{X}, \mathcal{Y}) \cup (\mathcal{X}', \mathcal{Y}') \cup (\mathcal{X}, \mathcal{Y}'')$.

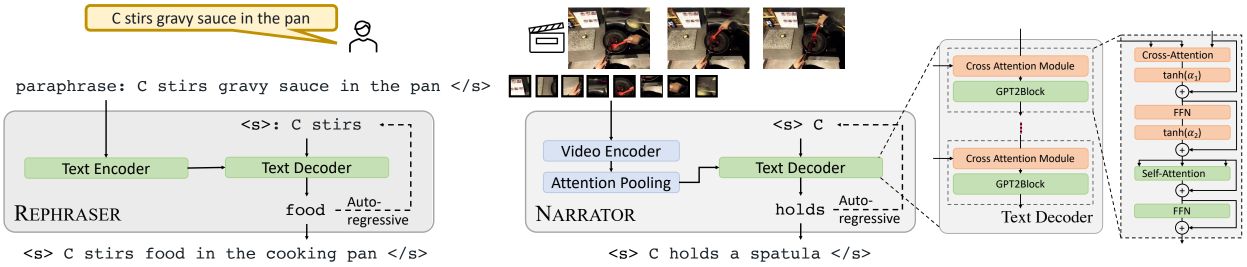


Figure 4. **Language supervision from REPHRASER and NARRATOR**. REPHRASER*(left)* takes the narration as input, passes it through a text encoder and uses a text decoder to autoregressively generate the rephrased output. NARRATOR*(right)* takes video frames as input and obtains the visual embeddings through a video encoder followed by attentional pooling. Equipped with a few additional cross-attention modules, the text decoder autoregressively generates new narrations for those new frames.

4.1. NARRATOR

Traditional LLMs, such as GPT-2 [50], are trained to generate a sequence of text tokens $(s_1 \cdots s_L)$ from scratch by modeling the probability of the next token given all tokens seen so far: $p(s_l | s_{<l})$. NARRATOR repurposes existing LLMs to be conditioned on the visual input and is trained on the original annotations $(\mathcal{X}, \mathcal{Y})$. The resulting model produces dense new annotations $(\mathcal{X}', \mathcal{Y}')$ on the full video. Following the formulation of factorized probabilities in language models [5], we model the visually conditioned text likelihood as follows:

$$p_{\text{NARRATOR}}(y'|x') = \prod_{\ell=1}^{L} p(s'_\ell | s'_{<\ell}, x'). \tag{2}$$

Architecture. We design NARRATOR to closely follow the architecture of standard LLMs, with only a few additional cross-attention modules added to provide visual conditioning, as illustrated in Figure 4 (*right*). This enables NARRATOR to be initialized from pre-trained weights, which is crucial for our task as the data we use to train NARRATOR (narrations associated with video clips) are far smaller in scale compared to the large text corpus typically used to train LLMs. Moreover, video narrations are less diverse and noisier because they are either collected by only a few annotators or automatically transcribed from speech. Similar "frozen-LM" approaches have shown effectiveness in multimodal few-shot adaptation in recent work [1, 65].

Specifically, we take a frozen pre-trained LLM and add a cross-attention module before each Transformer decoder layer so that the text input can attend to visual information. The cross-attended output then sums with the input text feature via residual connection [26] and goes to the Transformer decoder layer. Each cross-attention module comprises a cross-attention layer, which takes textual tokens as queries and visual embedding as keys and values, followed by a feed-forward network (FFN). Layer Normalization [3] is applied at the beginning of both cross-attention and FFN. We add `tanh`-gating [27], with an initial value of zero, such that the output of the new model is the same as that from the original language model at the beginning.

While features from any video model are applicable for conditioning, for convenience we adopt the video encoder from $\mathcal{F}$ in § 3, trained contrastively on the ground-truth data $(\mathcal{X}, \mathcal{Y})$. We use features before global pooling to allow the LLM to leverage fine-grained spatial-temporal information.

Training. We train NARRATOR on all of, or a subset of, the ground-truth annotations $(\mathcal{X}, \mathcal{Y})$. For each pair (x, y), the captioning loss is the sum of the negative log-likelihood of the correct word at each step,

$$\mathcal{L}_{\text{NARRATOR}}(x, y) = -\sum_{\ell=1}^{L} \log p(s_\ell | s_{<\ell}, x). \tag{3}$$

Inference. At inference time, we query NARRATOR by feeding visual input x plus a special start-of-sentence token `<s>`. We sample from the distribution recursively, *i.e.* $\tilde{s}_\ell \sim p(s | [\texttt{<s>}, \cdots, \tilde{s}_{\ell-1}], x)$ until an end-of-sentence token `</s>` is reached. Following [29], at each step we sample from a subset of tokens that contain the vast majority of the probability mass, which is known as nucleus sampling.

The effect of nucleus sampling is two-fold. On the one hand, it generates more diverse, open-ended, and human-like text than maximum-likelihood-based methods such as beam search and its variants [67]. On the other hand, the generated text may contain irrelevant or noisy information due to sampling without post-processing based on sentence-level likelihood. To address this, we repeat the sampling process for K times on the same visual input. We later demonstrate that the contrastive pre-training objective is robust to the noise caused by sampling, and the final performance benefits from a more diverse set of narrations.

To sample video clips for captioning, we start by simply re-captioning the existing clips labeled in the dataset $\mathcal{X}$, resulting in expanded annotations. Furthermore, long-form videos are typically sparsely narrated, meaning that the temporal union of all labeled clips cannot cover the entire video. Hence, we use NARRATOR to annotate the remainder of the video to obtain additional annotations by pseudo-captioning. With a simple assumption that video is a stationary process, we uniformly sample clips from the unlabeled intervals. The clip duration is equal to the average of all ground-truth clips, *i.e.* $\Delta = \frac{1}{N}\sum_{i=1}^{N}(e_i - t_i)$ while the sampling stride is computed likewise. Finally, by combining both re-captioned and pseudo-captioned narrations, we refer to the final set of annotations generated by Narrator as $(\mathcal{X}', \mathcal{Y}')$.

Post-processing. Exhaustive pseudo-captioning may contain some uninformative visual clips and generate text that is not useful. Thus, we add a filtering process to eliminate low-quality clips and their associated descriptions. We use the baseline dual-encoder model $\mathcal{F}$, which is trained on the ground-truth paired clips, to compute the visual and textual embedding of pseudo-labeled pairs and filter based on the similarity score, *i.e.* $\mathrm{Filter}(f_{\mathrm{v}}(x'_j)^\top \cdot f_{\mathrm{t}}(y'_j))$, where $\mathrm{Filter}(\cdot)$ can be either top-k of all generated text or a threshold filtering. In the experiments, we use a threshold of 0.5.

4.2. Rephraser

The data generated by Narrator is several times larger than the ground-truth pairs. To ensure that we do not overfit the pseudo-labeled data, we increase the number of ground-truth narrations by paraphrasing. In particular, we use a text-to-text LLM which models conditional text likelihood:

$$p_{\text{Rephraser}}(y''|y) = \prod_{\ell=1}^{L} p(s''_\ell | s''_{<\ell}, y).$$

The text-to-text model is implemented by an encoder-decoder architecture, *e.g.* T5 [51], to auto-regressively generate a new sentence given the original one. We observe that Rephraser is able to do basic manipulations such as replacing synonyms or changing word order, which serves as an efficient way of automatic data augmentation. The resulting annotations are referred to as $(\mathcal{X}, \mathcal{Y}'')$.

4.3. Training the Dual-Encoders

We train the dual-encoders as described in Algorithm 1 in Appendix E. In each iteration, we first sample a batch $\mathcal{B}$ of video clips. It comprises a subset of clips $\mathcal{B}_l$ with labeled timestamps as well as narrations, and a subset $\mathcal{B}_u$ whose clips are randomly sampled from videos without narrations. For clip $x_i \in \mathcal{B}_u$, we obtain the pseudo-caption y'_i by querying the Narrator $y'_i \sim p_{\text{Narrator}}(y'|x)$, resulting in a set of clips with LLM-generated narrations $\widetilde{\mathcal{B}}_u$. For clip $(x_i, y_i) \in \mathcal{B}_l$, the text supervision is obtained from either the Rephraser or the Narrator, with a probability of 0.5. We denote the resulting set of pairs to be $\widetilde{\mathcal{B}}_l$ similarly. Following CLIP [49], we use the symmetric cross-entropy loss over the similarity scores of samples in the batch $\widetilde{\mathcal{B}}_l \cup \widetilde{\mathcal{B}}_u$.

In practice, we run Rephraser and Narrator in advance and cache the resulting video-narration pairs so that there is no computational overhead during pre-training. Therefore, training a dual-encoder in LaViLa is as fast as training a standard dual-encoder contrastive model.

Datasets	Task	Ego?	Metrics	Eval. Prot.
EK-100 [14]	MIR	✓	mAP, nDCG	ZS, FT
EK-100 [14]	CLS	✓	top-1 action acc.	FT
Ego4D [24]	MCQ	✓	inter-/intra-video acc.	ZS
Ego4D [24]	NLQ	✓	Recall@N	FT
EGTEA [37]	CLS	✓	top-1, mean acc.	ZS, FT
CharadesEgo [58]	CLS	✓	video-level mAP	ZS, FT
UCF-101 [60]	CLS	✗	mean acc.	LP
HMDB-51[1] [35]	CLS	✗	mean acc.	LP

Table 1. **Downstream datasets** and metrics used for evaluation. We evaluate LaViLa on a wide range of tasks including Multi-Instance Retrieval (MIR), Multiple-Choice Question (MCQ), Natural Language Query (NLQ), and Action Recognition (CLS). The evaluation protocols include zero-shot (ZS), fine-tuning (FT), and linear-probing (LP). Please refer to Appendix C for more details.

5. Experiments

Dual-Encoder Architecture. The video-language model follows a dual-encoder architecture as CLIP [49]. The Visual encoder is a TimeSformer (TSF) [6], whose spatial attention modules are initialized from a ViT [18] which is contrastively pre-trained on large-scale paired image-text data as in CLIP [49]. We sample 4 frames per clip during pre-training and 16 when finetuning on downstream tasks. The text encoder is a 12-layer Transformer [50, 66]. We use BPE tokenizer [55] to pre-process the full sentence corresponding to the video clip and keep at most 77 tokens.

Narrator's architecture is a visually conditioned auto-regressive Language Model. The visual encoder is by default TimeSformer-L while the text decoder is a GPT-2 XL. During inference, we use nucleus sampling [29] with $p = 0.95$ and return $K = 10$ candidate outputs.

Rephraser. We use an open-source paraphraser [23] based on T5-large [51]. It is pre-trained on C4 [51] and then finetuned on a cleaned subset of ParaNMT [74]. During inference, we use Diverse Beam Search [67] with group number the same as beam number ($G = B = 20$) and set the diversity penalty to be 0.7. We keep 3 candidates per sentence, remove punctuations, and do basic de-duplication.

Pre-training dataset. We train on the video-narration pairs from Ego4D [13, 24], the largest egocentric video dataset to date. We exclude videos that appear in the validation and test sets of the Ego4D benchmark and determine each clip's interval using the same pairing strategy in [39]. This results in around 4M video-text pairs with an average clip length of 1 second. We also experiment with third-person videos by pre-training on HowTo100M [45] in § 5.2.

[1] HMDB data is licensed under the CC BY 4.0 license and the data is available at https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/

Method	Backbone	mAP			nDCG		
		V→T	T→V	Avg.	V→T	T→V	Avg.
(Zero-shot)							
EgoVLP [39]	TSF-B	19.4	13.9	16.6	24.1	22.0	23.1
EgoVLP* [39]	TSF-B	26.0	20.6	23.3	28.8	27.0	27.9
LaViLa	TSF-B	**35.1**	**26.6**	**30.9**	**33.7**	**30.4**	**32.0**
LaViLa	TSF-L	**40.0**	**32.2**	**36.1**	**36.1**	**33.2**	**34.6**
(Finetuned)							
MME [75]	TBN	43.0	34.0	38.5	50.1	46.9	48.5
JPoSE [75]	TBN	49.9	38.1	44.0	55.5	51.6	53.5
EgoVLP [39]	TSF-B	49.9	40.5	45.0	60.9	57.9	59.4
LaViLa	TSF-B	**55.2**	**45.7**	**50.5**	**66.5**	**63.4**	**65.0**
LaViLa	TSF-L	**54.7**	**47.1**	**50.9**	**68.1**	**64.9**	**66.5**

Table 2. **EK-100 MIR**. LaViLa outperforms prior work across all settings, metrics and directions of retrieval, with larger gains when switching to a larger model. Specifically, our best model achieves over 10% absolute gain in the zero-shot setting and 5.9 ∼ 7.1% gain in the finetuned setting. EgoVLP* refers to our improved version of [39], details of which are given in Appendix F.

Method	EgoMCQ		EgoNLQ			
	Accuracy (%)		mIOU@0.3		mIOU@0.5	
	Inter-video	Intra-video	R@1	R@5	R@1	R@5
SlowFast [24]	-	-	5.45	10.74	3.12	6.63
EgoVLP [39]	90.6	57.2	**10.84**	18.84	**6.81**	13.45
LaViLa (B)	**93.8**	**59.9**	10.53	**19.13**	6.69	**13.68**
LaViLa (L)	**94.5**	**63.1**	**12.05**	**22.38**	**7.43**	**15.44**

Table 3. **Ego4D EgoMCQ and EgoNLQ.** LaViLa outperforms prior work on both Multiple-Choice Questions and Natural Language Questions on Ego4D, with nearly 6% absolute gain on the challenging intra-video MCQ task that requires reasoning over multiple clips from the same video to answer a question.

Method	Backbone	Pretrain	Top-1 Acc.	Mean Acc.
Li *et al.* [37]	I3D	K400	-	53.30
LSTA [63]	ConvLSTM	IN-1k	61.86	53.00
IPL [71]	I3D	K400	-	60.15
MTCN [33]	SlowFast (V+A+T)	K400+VGG-Sound	73.59	65.87
Visual only	TSF-B	IN-21k+K400	65.58	59.32
LaViLa	TSF-B	WIT+Ego4D	**77.45**	**70.12**
LaViLa	TSF-L	WIT+Ego4D	**81.75**	**76.00**

Table 4. **EGTEA Classification**. LaViLa obtains significant gains on this task, outperforming prior work with over 10% mean accuracy. Since the backbones used are not all comparable, we also report a comparable baseline with TSF-B ("Visual only").

Evaluation protocols. We evaluate the learned video-text encoders using three evaluation protocols. (1) *Zero-Shot (ZS)*, meaning that we apply the pre-trained video-text encoders directly on the downstream validation dataset to perform video↔text retrieval tasks, without any tuning on the downstream dataset. Zero-shot classification is performed similarly, where we compute the similarity score between the video clip and the textual description of all possible categories. (2) *Finetuned (FT)*, where we take the pre-trained video-text model and perform end-to-end fine-tuning on the training split of the target downstream dataset. (3) *Linear-Probe (LP)*, where we compute the video features from a frozen encoder and train a linear SVM on top of the training split of the downstream dataset.

Downstream benchmarks. We use multiple benchmarks across four first-person (egocentric) and two third-person datasets, as enumerated in Table 1. We summarize them here and refer the reader to Appendix C for details on datasets and metrics. (1) Two tasks on Epic-Kitchens-100: Multi-Instance Retrieval (**EK-100 MIR**) and Action Recognition (**EK-100 CLS**) [14]. EK-100 is a very popular and challenging egocentric video recognition benchmark. The MIR task requires retrieving the text given videos (V→T) and videos given text (T→V). The CLS task requires classifying each video into one of 97 verbs and 300 nouns each, resulting in a combination of 3,806 action categories. (2) Two downstream tasks of Ego4D: Multiple-Choice Questions (**EgoMCQ**) and Natural Language Query (**EgoNLQ**). EgoMCQ requires selecting the correct textual description from five choices given a query video clip while EgoNLQ asks the model to output the relevant temporal intervals of video given a text query. We select these two benchmarks because they require reasoning about both visual and textual information. (3) Action Recognition on **EGTEA** [37]. It requires classifying into 106 classes of fine-grained cooking activities. (4) Action Recognition on **CharadesEgo** [58]. It requires classification into 157 classes of daily indoor activities. Note that CharadesEgo is very different from EK-100, Ego4D and EGTEA since its videos are captured by head-mounted phone cameras in a crowd-sourcing way.

In all tables, we bold and underline the best and second-best performing methods with comparable backbones architectures. We highlight the overall best performing method, which typically uses a larger backbone, if applicable.

5.1. Main Results

EK-100. We compare LaViLa with prior works on EK-100 MIR in Table 2. In the zero-shot setup, LaViLa remarkably surpasses an improved version of EgoVLP [39] under similar model complexity: we use TSF-Base+GPT-2 as the dual-encoder architecture while EgoVLP uses TSF-Base+Distil-BERT. With a stronger video encoder, *i.e.* TSF-Large, the performance improves further. In the fine-tuned setting, LaViLa significantly outperforms all previous supervised approaches including MME, JPOSE [75] and EgoVLP [39]. We also compare LaViLa on EK-100 CLS in Appendix E, and establish a new state-of-the-art.

Ego4D. We evaluate the pre-trained LaViLa model on EgoMCQ and EgoNLQ tasks and compare the results in Table 3. On EgoMCQ, our method achieves 93.8% inter-video accuracy and 59.9% intra-video accuracy, outperforming EgoVLP by a noticeable margin. Note that EgoVLP's per-

Method	Backbone	mAP (ZS)	mAP (FT)
ActorObserverNet [57]	ResNet-152	-	20.0
SSDA [12]	I3D	-	25.8
Ego-Exo [38]	SlowFast-R101	-	30.1
EgoVLP [39]	TSF-B	25.0	32.1
LAVILA	TSF-B	**26.8**	**33.7**
LAVILA	TSF-L	**28.9**	**36.1**

Table 5. **CharadesEgo Action Recognition**. LAVILA sets new state-of-the-art in both zero-shot (ZS) and finetuned (FT) settings. Note that CharadesEgo videos are visually different compared to Ego4D videos, on which LAVILA is pretrained.

formance reported in Table 3 is obtained by using EgoNCE loss [39], a variant of InfoNCE specialized for Ego4D while ours uses a standard InfoNCE loss. EgoVLP with InfoNCE has lower performance (89.4% inter-video and 51.5% intra-video accuracy). On EgoNLQ, LAVILA achieves comparable results with EgoVLP with similar model complexity.

EGTEA. We evaluate the learned video representation by finetuning the video encoder for action classification in Table 4 on another popular egocentric dataset, EGTEA [37]. Our method surpasses the previous state-of-the-art which takes multiple modalities including visual, auditory and textual inputs [33] by a more than 10% absolute margin on the mean accuracy metric. Since previous methods are based on different backbones, we experiment with a TSF-Base ("Visual only") model pre-trained on Kinetics [9] as a fair baseline for LAVILA. We observe that its accuracy is comparable to previous methods but much lower than LAVILA, implying the effectiveness of learning visual representation on large-scale egocentric videos and using LLM as textual supervision during pre-training.

CharadesEgo. Next, we compare LAVILA's representation on the CharadesEgo action classification task. As shown in Table 5, LAVILA's representation excels on this task as well, which is notable as CharadesEgo videos are significantly different compared to Ego4D, being captured by crowdsourced workers using mobile cameras.

Method	Vis. Enc.	UCF-101	HMDB-51
MIL-NCE [44]	S3D	82.7	54.3
TAN [25]	S3D	83.2	56.7
Baseline (w/o LLM)	TSF-B	86.5	**59.4**
LAVILA	TSF-B	**87.4**	57.2
LAVILA	TSF-L	**88.1**	**61.5**

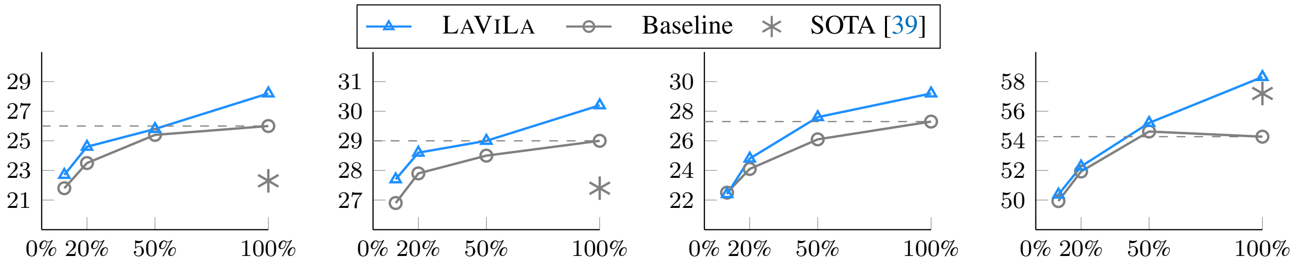
Table 6. **LAVILA on third-person videos**. We measure the linear-probing action classification performance of the video model after pre-training on HowTo100M [45].

5.2. Application to Third-Person Video Pre-training

We apply LAVILA to third-person videos by experimenting with the HowTo100M [45] dataset. Specifically, we use the temporally aligned subset provided by [25], which contains 3.3M sentences from 247k videos. We evaluate the video representation on two third-person video datasets, *i.e*. UCF-101 [60] and HMDB-51 [35] for action classification using the linear probing protocol. For more details, please refer to Appendix D. From Table 6, we see that LAVILA outperforms previous methods such as MIL-NCE [44] and TAN [25] by a large margin. Since we use a different backbone, we report a baseline without LLM and show that LAVILA indeed benefits from the language supervision.

5.3. Application to Semi-supervised Learning

While LAVILA is very effective at leveraging existing narrations to augment them, we now show that it is also applicable when only a limited number of narrations are available to begin with. We first divide each long video from Ego4D into 15-second chunks and assume only the annotated clips within every N chunks is available during pre-training, leading to approximately $\frac{100}{N}\%$ of the full set, where $N \in \{2, 5, 10\}$. This can be considered a practical scenario when we want to annotate as many videos as possible for diversity when the annotation budget is limited. In the remainder $(1 - \frac{100}{N}\%)$ part that is skipped, we uniformly sample the same number of the clips per chunk with the same clip length as that in the seen chunks. Both



(a) EK-100 MIR mAP. (b) EK-100 MIR nDCG. (c) EGTEA mean accuracy. (d) EgoMCQ Intra-video accuracy.

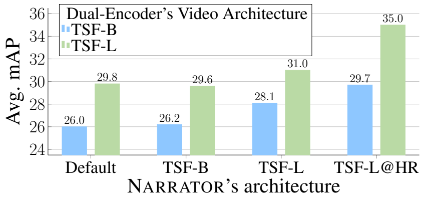
Figure 5. **LAVILA is effective in a semi-supervised setting where only a limited amout of narrations are given**. Comparing zero-shot performance of pre-training, LAVILA consistently outperforms the groundtruth-only baseline when 10, 20, 50, 100% data is used. We also achieve comparable result with state-of-the-art with only 50% of the annotated data.

Text Dec. arch.	Text Dec. init.	Freeze LM	METEOR	ROUGE-L	CIDEr	Avg. mAP
(baseline)			-	-	-	26.0
GPT-2	random	✗	0.284	0.524	0.882	24.3
GPT-2	WebText	✓	0.270	0.505	0.804	24.0
GPT-2 XL	WebText	✓	**0.289**	**0.530**	**0.940**	**26.2**

(a) **Generation Quality**. Using a sufficiently large language model as the text decoder is crucial for good text generation quality and downstream performance.

Sampling method	# of sentences	Avg. mAP
N/A (baseline)	-	26.0
Beam search	1	27.9
Nucleus	1	29.6
Nucleus	10	**29.7**

(b) **Sampling**. LAVILA benefits more from narrations produced by nucleus sampling than beam search.



(c) **Scaling effect of LAVILA.** Gains increase on scaling the video encoder in NARRATOR. Default refers to only using the original narrations.

Table 7. **Ablations of NARRATOR**. We report zero-shot average mAP on EK-100 MIR for comparing downstream performance. We study NARRATOR from the perspective of generation quality (*left*), sampling techniques (*middle*), and scaling effect (*right*).

Rephr.	Recap.	Pseudo cap.	EK-100 MIR		EgoMCQ		EGTEA	
			Avg. mAP	Avg. nDCG	inter-video	Intra-video	Mean	Top-1
			26.0	28.8	**93.6**	54.3	27.3	30.1
✓			28.0	30.1	93.5	56.9	<u>29.8</u>	30.8
	✓		27.1	29.9	93.2	**59.2**	26.8	31.2
✓	✓		<u>29.7</u>	**31.5**	**93.6**	58.3	29.4	**36.6**
✓	✓	✓	**29.9**	<u>31.4</u>	**93.6**	<u>59.1</u>	**31.1**	<u>36.0</u>

Table 8. **Contributions of different Language Supervision.** We can see that (1) using REPHRASER ("Rephr.") and NARRATOR ("Recap.") improve downstream zero-shot performance complementarily, (2) dense pseudo-captioning further improves performance on 3 out of 6 metrics.

the dual-encoder model and NARRATOR are trained on the $\frac{100}{N}$% available annotations.

We plot the zero-shot performance curve of pre-training with different proportions in Figure 5. We can see that LAVILA consistently outperforms the ground-truth-only baseline at all points (10, 20, 50, and 100%). The improvement tends to be larger when more data is available, indicating the method's scalability as more videos are narrated in the future. Furthermore, we observe our method can achieve a similar level of performance with the baseline often using less than 50% data. We also achieve a comparable result with the state-of-the-art using much fewer data.

5.4. Ablation Studies

Contributions of Different Language Supervisions. We ablate different language supervisions in Table 8 on EK-100 MIR (zero-shot), EgoMCQ and EGTEA. Using the text-only REPHRASER ("rephr.") or visually conditioned NARRATOR ("recap.") separately improves the ground-truth baseline noticeably. Combining both REPHRASER and NARRATOR gives an improvement of 3.5% average mAP on EK-100 MIR. We see that dense captioning on the entire video ("pseudo-cap.") is also helpful. Though the gain on EK-100 MIR is not as significant, it shows nontrivial improvements on EgoMCQ intra-video accuracy and EGTEA mean accuracy. Our conjecture for this marginal gain is that informative clips are mostly covered in Ego4D because all videos are inspected by two annotators.

Generation Quality of NARRATOR. We study how the NARRATOR's configurations affect the quality of generated text and the downstream performance. The generation quality is measured by standard unsupervised automatic metrics including METEOR, ROUGE, and CIDEr [43]. We use a NARRATOR with a smaller GPT-2 as the text decoder and consider two scenarios in Table 7a: (1) LM is randomly initialized but jointly trained with the gated cross-attention modules, and (2) LM is initialized from the original GPT-2. The generation quality decreases compared to GPT-2 XL in both cases and the zero-shot retrieval result on EK-100 MIR is worse. This indicates that the language model should be sufficiently large and pre-trained on web text data.

Sampling. In Table 7b, we investigate different sampling methods for text generation from NARRATOR. We see that nucleus sampling works much better than beam search while repetitive sampling shows marginal improvement.

Scaling effect. In Table 7c, we compare the zero-shot retrieval result by progressively increasing the size of NARRATOR's video encoder from TSF-B to TSF-L and TSF-L@HR, which increases the input resolution to be narrated from 224 to 336 while fixing the dual-encoder architecture. The retrieval performance steadily increases while NARRATOR becomes stronger. We conduct this experiment by varying the dual-encoder architecture, namely TSF-Base and TSF-Large, and show similar trends. Both phenomena suggest that LAVILA can scale to larger models.

6. Conclusion and Future Work

In this paper, we proposed LAVILA, a new approach to video-language representation learning by automatically narrating long videos with LLMs. We achieve strong improvements over baselines trained with the same amount of human-narrated videos and set new state-of-the-art on six popular benchmark tasks across first- and third-person video understanding benchmarks. LAVILA also shows positive scaling behavior when adding more training narrations, using larger visual backbones, and using stronger LLMs, all of which are promising areas for future work.

Acknowledgements: We thank Naman Goyal, Stephen Roller and Susan Zhang for help with language models, Kevin Qinghong Lin for help with EgoVLP, and the Meta AI team for helpful discussions and feedback. 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, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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

[5] Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. In *NIPS*, 2000.

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

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *NeurIPS*, 2020.

[8] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

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

[10] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

[11] Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. Unifying vision-and-language tasks via text generation. In *ICML*, 2021.

[12] Jinwoo Choi, Gaurav Sharma, Manmohan Chandraker, and Jia-Bin Huang. Unsupervised and semi-supervised domain adaptation for action recognition from drones. In *WACV*, 2020.

[13] Ego4D Consortium. Egocentric live 4d perception (Ego4D) database: A large-scale first-person video database, supporting research in multi-modal machine perception for daily life activity. https://sites.google.com/view/ego4d/home. Accessed: 2022-11-22.

[14] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: collection, pipeline and challenges for Epic-Kitchens-100. *IJCV*, 2022.

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[16] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. In *CVPR*, 2021.

[17] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020.

[19] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. In *ACL*, 2017.

[20] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. In *ACL Findings*, 2021.

[21] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. *NeurIPS*, 2013.

[22] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *CVPR*, 2022.

[23] Ramsri Goutham Golla. High-quality sentence paraphraser using transformers in nlp. https://huggingface.co/ramsrigouthamg/t5-large-paraphraser-diverse-high-quality. Accessed: 2022-06-01.

[24] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will

Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

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

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

[27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[28] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

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

[30] Xiaowei Hu, Zhe Gan, Jianfeng Wang, Zhengyuan Yang, Zicheng Liu, Yumao Lu, and Lijuan Wang. Scaling up vision-language pre-training for image captioning. In *CVPR*, 2022.

[31] 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.

[32] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, 2022.

[33] Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen. With a little help from my temporal context: Multimodal egocentric action recognition. In *BMVC*, 2021.

[34] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *CVPR*, 2021.

[35] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[36] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *CVPR*, 2021.

[37] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *ECCV*, 2018.

[38] Yanghao Li, Tushar Nagarajan, Bo Xiong, and Kristen Grauman. Ego-exo: Transferring visual representations from third-person to first-person videos. In *CVPR*, 2021.

[39] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Wenzhe Zhao, Weijie Kong, Hongfa Cai Chengfei, Wang, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pretraining. In *NeurIPS*, 2022.

[40] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *ECCV*, 2022.

[41] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *NeurIPS*, 2019.

[42] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020.

[43] Maluuba. nlg-eval. `https://github.com/Maluuba/nlg-eval`. Accessed: 2022-06-01.

[44] 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 *CVPR*, 2020.

[45] 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.

[46] 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.

[47] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *ECCV*, 2022.

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

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

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

[51] 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.

[52] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[53] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville,

and Bernt Schiele. Movie description. *IJCV*, 2017.

[54] Christoph Schuhmann, Romain Beaumont, Cade W Gordon, Ross Wightman, Theo Coombes, Aarush Katta, Clayton Mullis, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS D&B*, 2022.

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

[56] Paul Hongsuck Seo, Arsha Nagrani, Anurag Arnab, and Cordelia Schmid. End-to-end generative pretraining for multimodal video captioning. In *CVPR*, 2022.

[57] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *CVPR*, 2018.

[58] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Charades-ego: A large-scale dataset of paired third and first person videos. *arXiv preprint arXiv:1804.09626*, 2018.

[59] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *NeurIPS*, 2016.

[60] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[61] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *TMLR*, 2022.

[62] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020.

[63] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *CVPR*, 2019.

[64] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019.

[65] Maria Tsimpoukelli, Jacob L Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *NeurIPS*, 2021.

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[67] Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

[68] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[69] 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 all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*, 2022.

[70] William Yang Wang and Diyi Yang. That's so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *EMNLP*, 2015.

[71] Xiaohan Wang, Linchao Zhu, Heng Wang, and Yi Yang. Interactive prototype learning for egocentric action recognition. In *ICCV*, 2021.

[72] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP*, 2019.

[73] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 2010.

[74] John Wieting and Kevin Gimpel. Paranmt-50m: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *ACL*, 2018.

[75] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, 2019.

[76] 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 *CVPR*, 2022.

[77] 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.

[78] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. *arXiv preprint arXiv:2209.06430*, 2022.

[79] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, 2022.

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

[81] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[82] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *NeurIPS*, 2021.

[83] Andy Zeng, Adrian Wong, Stefan Welker, Krzysztof Choromanski, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv preprint arXiv:2204.00598*, 2022.

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

[85] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NeurIPS*, 2015.

[86] Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards

automatic learning of procedures from web instructional videos. In *AAAI*, 2018.

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

[88] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.

[89] Xizhou Zhu, Jinguo Zhu, Hao Li, Xiaoshi Wu, Hongsheng Li, Xiaohua Wang, and Jifeng Dai. Uni-perceiver: Pre-training unified architecture for generic perception for zero-shot and few-shot tasks. In *CVPR*, 2022.

A. Radar Chart Figure 1 Details

We first describe how we plot the radar chart in Figure 1. Each axis denotes a specific metric on one video understanding task. Each vertex denotes a ratio relative to our performance, which is computed by normalizing the performance of either LaViLa or previous SOTA by that of LaViLa, and is in the range of $(0, 1]$. For illustrative purpose, we set the radar chart's origin to be 80% and outermost frame to be 100% so that the interval between neighboring lattices to be 5%. The numbers annotated next to the vertices are *absolute value* of performance *without normalization*. Note that in other radar charts [69, 80], the axes have different scales and interval values while the origin is not valid, which may lead to potential fallacies.

B. LaViLa Details

The algorithm of training LaViLa is given in Algorithm 1. The loss is based on the CLIP [49]'s symmetric cross-entropy loss over the similarity scores of samples in the batch $\widetilde{\mathcal{B}}_l \cup \widetilde{\mathcal{B}}_u$ with minimal modifications. We apply two separate temperatures (τ_r, τ_n) for embeddings from rephrased pairs and pseudo-captioned ones respectively,

$$\mathcal{L} = -\frac{1}{2N}\sum_{i=1}^{N}\left(\log\frac{\exp(\frac{\mathbf{v}_i^\top \mathbf{u}_i}{\tau_i})}{\sum\limits_{j=1}^{N}\exp(\frac{\mathbf{v}_i^\top \mathbf{u}_j}{\sqrt{\tau_i\tau_j}})} + \log\frac{\exp(\frac{\mathbf{u}_i^\top \mathbf{v}_i}{\tau_i})}{\sum\limits_{j=1}^{N}\exp(\frac{\mathbf{u}_i^\top \mathbf{v}_j}{\sqrt{\tau_i\tau_j}})}\right). \tag{4}$$

We ablate different choices of temperatures in Table 12.

Algorithm 1 One step of training LaViLa

```
Require: A subset of narrated (unnarrated) clips $\mathcal{B}_l$ ($\mathcal{B}_u$)
  clips with LM-generated narrations $\widetilde{\mathcal{B}}_l = \{\}, \widetilde{\mathcal{B}}_u = \{\}$
  for $(x_i, y_i) \in \mathcal{B}_l$ do
      $u \sim U(0, 1)$        ▷ Uniform sample between 0 and 1
      if $u < 0.5$ then        ▷ Query Rephraser
          $y'_i \sim p_{\text{Rephraser}}(y'|y_i), \tau_i \leftarrow \tau_r$
      else        ▷ Query Narrator
          $y'_i \sim p_{\text{Narrator}}(y'|x_i), \tau_i \leftarrow \tau_n$
      end if
      $\widetilde{\mathcal{B}}_l \leftarrow \widetilde{\mathcal{B}}_l \cup \{(x_i, y'_i, \tau_i)\}$
  end for
  for $x_i \in \mathcal{B}_u$ do
      $y'_j \sim p_{\text{Narrator}}(y'|x_i), \tau_j \leftarrow \tau_n$
      $\widetilde{\mathcal{B}}_u \leftarrow \widetilde{\mathcal{B}}_u \cup \{(x_j, y'_j, \tau_j)\}$
  end for
  Train $\mathcal{F}_{\text{LaViLa}}(x, y)$ with the batch $\widetilde{\mathcal{B}}_l \cup \widetilde{\mathcal{B}}_u$ using Eq 4.
```

C. Dataset Details

In this section, we provide details of the datasets where we conduct experiments.

Ego4D. Ego4D contains 3,670 hours of egocentric videos with temporally dense narrations. Each narration has a timestamp and an associated free-form sentence. We construct the video-text clip pairs that are used for pre-training following [39]. First, we exclude 2,429 videos that appear in the validation and test sets of the Ego4D benchmark. Next, we determine the each clip's interval using the contextual variable-length clip pairing strategy in [39]. Finally, we drop the narrations that either contain "#unsure"/"#Unsure" tags or are shorter than 4 words. This results in 4,012,853 video-text clip pairs with an average clip length of $1(\pm 0.9)$ second. For the excluded videos, we also pre-process similarly and obtain 1,260,434 video-text clip pairs. We only use them as validation split to measure the generation quality of Narrator in Table 7a.

EK-100. The Epic-Kitchens-100 (EK-100) dataset contains 100 hours of egocentric cooking videos. The training split has 67,217 video clips; the validation split has 9,668 video clips; the testing split has 13,092 video clips. Each clip is annotated with (1) a start and end timestamp, (2) a short textual narration, and (3) a verb and noun class that the narration belongs to. The action class can also be uniquely determined by combining the verb and the noun. In the zero-shot setting, we evaluate the pre-trained model on the validation split directly without any tuning; In the finetuned setting, we take the pre-trained model and perform end-to-end finetuning on the training split and evaluate on the validation split. For EK-100 MIR we use the textual narration while for EK-100 CLS we use the class of verb, noun, and action as the label. For EK-100 MIR, the evluation metrics are mean Average Precision (mAP) and normalized Discounted Cumulative Gain (nDCG). For EK-100 CLS, the evaluation metrics are top-1 accuracies for verb, noun, and action. Action-level accuracy is the most important one among all.

EGTEA. EGTEA contains 28 hours of egocentric cooking videos with gazing tracking. In our experiments, we take as input the visual frames only. The action annotations include 10,321 instances of fine-grained actions from 106 classes, with an average duration of 3.2 seconds. In the zero-shot setting, we evaluate the pre-trained model on the test set of all three splits without any tuning and report results as the mean accuracy averaged across all classes across all three splits, as Li *et al.* [37] suggested. In the finetuned setting, we follow prior works [33] and report top-1 accuracy and mean class accuracy using the first train/test split, which has 8,299/2,022 instances respectively.

CharadesEgo. The CharadesEgo dataset contains 7,860 videos of daily indoor activities from both third- and first-person views. The annotations are 68,536 instances of fine-grained actions from 157 classes. We use the first-person subset only, comprising 3,085 videos for training and 846 videos for testing. We report video-level mAP as the evaluation metric. In the zero-shot setting, we evaluate the pre-trained model on the test videos directly without any tuning; In the finetuned setting, we perform end-to-end finetuning on the trimmed action instances in the training split, which has an amount of 33,114 action instances.

D. Implementation Details

D.1. Pre-training on Ego4D

We pre-train on the video-narration pairs from Ego4D [24]. We train the model using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$ and weight decay of 0.01 for 5 epochs. We use a fixed learning rate of 3e-5. The projection head after the dual-encoders is a linear layer with an output dimension of 256. We use PyTorch's native FP16 mixed precision training and gradient checkpoint. This allows us to afford a per-gpu batch size of 32 over 32 GPUs for TimeSformer-B and a per-gpu batch size of 16 over 64 GPUs for TimeSformer-L, resulting in a total batch size of 1,024. We abate these design choices in Appendix F.

For input, we first divide each video into 5-minute segments and scale the short side of the video to 288 pixels. This signifantly reduces storage and accelerates decoding. During training, we decode the corresponding segment that contains the selected clip. We randomly sample 4 frames between the start and end time of the clip and use standard `RandomResizedCrop (0.5, 1.0)` for data augmentation.

D.2. Training Narrator on Ego4D

Architecture. For the video encoder, we use the one we obtain in Appendix D.1 and keep it frozen. We drop the global average pooling layer and attach an attention pooling module, which is instantiated by a standard cross-attention [66] and a Layer Normalization [3]. The attention pooling uses a fixed length of randomly initalized queries $\mathbf{q} \in \mathbb{R}^{N_\mathrm{q} \times D_\mathrm{t}}$ to attend visual features $\mathbf{v} \in \mathbb{R}^{(T \times H' \times W') \times D_\mathrm{v}}$. This results in a fixed length of hidden states, $\mathrm{AttentionPool}(\mathbf{q}, \mathbf{v}) \in \mathbb{R}^{N_\mathrm{q} \times D_\mathrm{t}}$, which will be later fed into the cross-attention module of the text decoder. This ensures the text decoder attends to the same number of visual features irrespective of the input visual resolution, *e.g.* 224×224 or 336×336. More concretely, $\mathrm{AttentionPool}(\mathbf{q}, \mathbf{v})$ is computed as follows:

$$\mathbf{q}', \mathbf{v}' = \mathrm{LayerNorm}(\mathbf{q}), \mathrm{LayerNorm}(\mathbf{v}),$$
$$\mathrm{head}_i = \mathrm{softmax}\left(\frac{(\mathbf{q}'\mathbf{W}_Q^{(i)})(\mathbf{v}'\mathbf{W}_K)^\top}{\sqrt{d_0}}\right) \cdot (\mathbf{v}'\mathbf{W}_V),$$
$$\mathrm{AttentionPool} = \mathrm{Concat}(\mathrm{head}_1, \cdots, \mathrm{head}_h) \cdot \mathbf{W}_O,$$

where $\mathbf{W}_Q \in \mathbb{R}^{D_\mathrm{t} \times d_0}$, $\mathbf{W}_{K/V} \in \mathbb{R}^{D_\mathrm{v} \times d_0}$, and $\mathbf{W}_O \in \mathbb{R}^{(h \cdot d_0) \times D_\mathrm{t}}$.

For the text decoder, we use GPT-2 XL [50] and keep it frozen. The video encoder and the text decoder is bridged by a cross-attention module. Each cross-attention module comprises a cross-attention layer followed by a feed-forward network (FFN). Layer Normalization is added at the beginning of both cross-attention and FFN. We add `tanh`-gating [27] with an initial value of zero. We insert one cross-attention module every two GPT2-Blocks in GPT2 XL to save memory. Both the attention pooling and cross-attention modules are learnable parameters.

We train Narrator on the ground-truth video-narration pairs from Ego4D [24]. The training recipe mostly follows the one for pre-training the dual-encoders except that we use FP32 to train Narrator because PyTorch's native FP16 mixed-precision leads to training instability. We use the video-text clip pairs from the Ego4D's validation videos to compute the word-level classification accuracy and perplexity. We select the model with the highest accuracy as well as lowest perplexity, which is often reached after 3∼4 epochs.

D.3. Multi-Instance Retrieval on EK-100

We fine-tune the pre-trained model on EK100 using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$ and weight decay of 0.01. We use cosine annealing with warmup, where the base learning rate starts from 1e-6, linearly increases to a peak of 3e-3 in the first epoch and then gradually decreases to 1e-5 following a half-wave cosine schedule. We apply the multi-instance max-margin loss [75] with a margin value of 0.2. We use a per-gpu batch size of 16 over 8 GPUs for TimeSformer-B and a per-gpu batch size of 4 over 32 GPUs for TimeSformer-L. We use a stochastic depth ratio of 0.1 in the backbone.

For the input, we represent each video clip with 16 sampled frames at both training and testing time. At training time, We scale the short side of the video to 256 pixels and then take a 224×224 crop while at testing time, we scale the short side to 224 pixels and take the center 224×224 crop.

D.4. Action Recognition on EGTEA

We fine-tune the pre-trained model on EGTEA for 100 epochs using SGD with a momentum of 0.9 and weight decay of 5e-4. We use cosine annealing with warmup, where the base learning rate starts from 1e-6, linearly increases to a peak of 3e-3 in the first epoch and then gradually decreases to 1e-5 following a half-wave cosine schedule. We drop the linear projection head and attach a 106-dim head for classification. For LaViLa, we train the classification head with 1× base learning rate and the backbone with $0.1\times$. For visual-only video model pre-trained on Kinetics, we use 1× base learning rate for both the classification head and the backbone. We use a per-gpu batch size of 16 over 8 GPUs for TimeSformer-B and a per-gpu batch size of 4 over 32 GPUs for TimeSformer-L. We use a stochastic depth ratio of 0.1 in the backbone and a dropout of 0.5 before the clas-

Method (Backbone)	Pretrain	Top-1 accuracy		
		Verb	Noun	Action
IPL (I3D) [71]	K400	68.6	51.2	41.0
ViViT-L [2]	IN-21k+K400	66.4	56.8	44.0
MoViNet [34]	N/A	**72.2**	57.3	47.7
MTV [79]	WTS-60M	69.9	**63.9**	<u>50.5</u>
MTCN (MFormer-HR) [33]	IN-21k+K400 +VGG-Sound	70.7	62.1	49.6
Omnivore (Swin-B) [22]	IN21k+IN-1k +K400+SUN	69.5	61.7	49.9
MeMViT [76]	K600	71.4	60.3	48.4
LaViLa (TSF-L)	WIT+Ego4D	<u>72.0</u>	<u>62.9</u>	**51.0**

Table 9. **The performance of action recognition on EK-100**. We report top-1 accuracy on verb, noun, and action. LaViLa outperforms all prior works in terms of action-level top-1 accuracy.

sification head. We also use a label smoothing of 0.1.

For input, we randomly select a 32-frame video clip at a temporal stride of 2 (namely 16×2) from each video at training time. We scale the short side of the video to 256 pixels and then take a 224×224 crop. For data augmentation, we use standard `RandomResizedCrop (0.5, 1.0)` and `RandomHorizontalFlip(0.5)`. At testing time, we evenly take ten 32-frame clips through the full video. We scale the short side to 224 pixels and take three spatial crops along the longer axis per clip. The final predictions are averaged over all these crops.

D.5. Action Recognition on EK-100

We fine-tune the pre-trained model on EK100 with a same training schedule as in EGTEA. The only exception is that we apply three classification heads for verb, noun, and action separately because we empirically observe that it speeds up convergence and performs slightly better than using a single action-level classification head.

For the input, we represent each video clip with 16 sampled frames at both training and testing time. At testing time, we take three spatial crops along the longer axis per clip and average the final predictions.

D.6. Action Recognition on CharadesEgo

Following EgoVLP [39], we convert the task of action classification to that of video-text retrieval as follows: for each trimmed video clip with textual annotations, we consider it to be a valid video-text pair for training. Since CharadesEgo is a multi-class dataset, which means each trimmed video can be annotated with different classes, we treat any trimmed video clip with N actions as N individual video-text pairs. We use the same InfoNCE [48] loss. We fine-tune the pre-trained model on CharadesEgo using AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$ and weight decay of 0.01. We use cosine annealing with warmup, where the peak learning rate is set to be 3e-5. For input, we randomly select a 32-frame video clip at a stride of 2 from the *trimmed* video at training time and evenly sample 16 frames from the *untrimmed* video at testing time to calculate the video-level mAP. We finetune the model for 10 epochs and report the best performance.

	EgoNCE	CLIP-init.	# frames	Avg. mAP	Avg. nDCG
Extracted RGB frames					
EgoVLP [39]			4	15.5	22.1
EgoVLP [39]	✓		4	16.6	23.1
Videos (downsized to 480p)					
EgoVLP [39]	✓		4	22.3	27.4
EgoVLP [39]	✓		16	23.6	27.9
Our impl.			4	24.1	28.0
Our impl.		✓	4	24.7	28.4

Table 10. **Improved baseline** evaluted on EK-100 MIR. We observe that evaluting on videos directly improves the baseline noticeably. Using CLIP-pre-trained encoder weights introduces additional improvements. All gains shown in the paper are on top of this already strong baseline (last row).

D.7. LaViLa for Third-person Video Pre-training

The pre-training recipe mostly follows the one in Appendix D.1 except that when constructing a batch of samples, we sample one more hard negative clip from the same video for each selected clip following [25].

When doing linear-probing evaluation, we keep the video encoder frozen, extract video feature and train a linear SVM on top. For each video clip in either HMDB-51 or UCF-101, we evenly take four 32-frame clips through the entire video. We scale the short side to 224 pixels and take the center crop per clip and pass through the frozen video encoder to get the final visual embedding. For each testing video, we average the prediction score from different clips. We use scikit-learn's LinearSVC and report the highest top-1 accuracy after sweeping the regularization parameter $C \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10^2, 10^3, 10^4\}$.

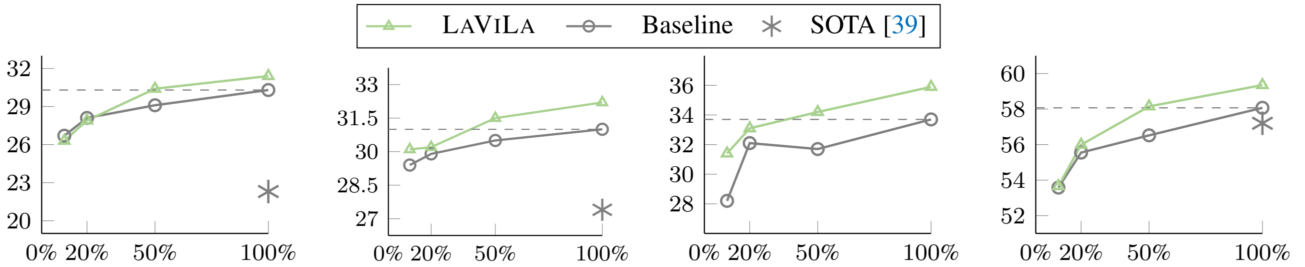
E. Additional Results

EK-100 CLS. We compare LaViLa representation on EK-100 CLS in Table 9. We achieve state-of-the-art performance in terms of top-1 action accuracy. Note that the second best-performing Multiview Transformer [79] is pre-trained on WTS-60M which is not publicly available.

More results on Semi-supervised Learning. Following the setup in § 5.3, we provide more results in Figure 6 while replacing the backbone of LaViLa with TimeSformer-Large. We observe similar trends as § 5.3 where LaViLa outperforms the ground-truth-only baseline at all data points.

F. Additional Ablations

Improved Baseline on EK-100 MIR. We present an improved baseline of video-language model pretrained on



(a) EK-100 MIR mAP. (b) EK-100 MIR nDCG. (c) EGTEA mean accuracy. (d) EgoMCQ Intra-video accuracy.

Figure 6. **More results of LaViLa in a semi-supervised setting where only a limited amount of narrations are given**. Both LaViLa and the baseline use a TimeSformer-Large as the visual encoder backbone. Comparing zero-shot performance of pre-training, LaViLa consistently outperforms the groundtruth-only baseline when 10, 20, 50,100% data is used.

Vis. Enc. arch.	Vis. Enc. init.	Text Enc.	Text Enc. init.	avg mAP	avg. nDCG
TSF-B	IN-21K	DistilBERT	BC+Wiki	24.1	28.0
TSF-B	WIT	DistilBERT	BC+Wiki	24.2	**28.5**
ViT-B	WIT	CLIP-GPT	WIT	23.2	27.4
TSF-B	WIT	CLIP-GPT	WIT	**24.7**	28.4

(a) **initialization**. IN-21K and WIT denote ImageNet-21k [15] and WebImageText [49]. BC+Wiki denotes BookCorpus+English Wikipedia on which BERT is pre-trained. Using CLIP-initialized weights works better than using those supervised pretrained on IK-21K.

Batch size	Avg. mAP	Avg. nDCG
512	24.7	28.4
1024	**25.6**	**28.8**
2048	**25.6**	28.5

(b) **Batch size**. Zero-shot performance improves when batch size increases from 512 to 1,024.

Projection head	Avg. mAP	Avg. nDCG
Linear (128-d)	24.1	27.8
Linear (256-d)	**24.7**	**28.4**
Linear (512-d)	24.5	28.1

(c) **Projection head**. Zero-shot performance is affected by the hidden dimension of the projection head. Empirically using 256 yields a best performance.

Table 11. **Ablations of dual-encoder**. We study how weight initialization (a), pre-training batch size (b), and project head dimension (c) affect the zero-shot performance of the dual-encoder on EK-100 MIR.

Ego4D and evaluate it on EK-100 MIR in a zero-shot setting in Table 10. The initial baseline is video-language model with a TimeSformer-Base as visual encoder and a DistilBERT as textual encoder, proposed in EgoVLP [39]. First, we find that zero-shot evaluation on videos brings a noticeable improvement than on extracted RGB frames. Particularly, given the same EgoVLP+EgoNCE model, zero-shot retrieval can increase by 5.7% average mAP and 4.3% average nDCG repespectively. This is probably because frame extraction using ffmpeg's default parameter downgrades the image quality by a considerable amount. Second, under the same video-as-input evaluation protocol, our implementation with the same backbone (TimeSformer-Base + DistilBERT) using standard InfoNCE loss *without* EgoNCE, can achieve 24.1% and 28.0% average mAP and nDCG, better than the EgoVLP with EgoNCE. Third, if we pretrain the joint model using CLIP-pretrained models as the initial weights, the zero-shot retrieval result can be further boosted (+0.6% avg. mAP and +0.4% avg. nDCG), indicating that egocentric video representation can also benefit from large-scale image-text pre-training.

Starting from this improved baseline, we conduct more ablations on pretraining the video-langauge model in Table 11 as follows. We measure the performance by zero-shot average mAP and average nDCG on EK-100 MIR.

Effect of weight initialization. We study the effect of architectures and weight initialization in Table 11a. First, we observe that using the same architecture of TimeSformer-B, using CLIP-initialized weights pretrained on WebImage-Text (WIT) [49] works slightly better than using those supervised pretrained on ImageNet-21k [15, 61]. Second, if we replace the visual encoder with a ViT-Base model as in CLIP, the performance drops by 1.5% avg. mAP and 1.0% avg. nDCG, indicating the necessity of using spatial-temporal visual encoder for learning video-language tasks.

Effect of batch size. We study the effect of batch size of contrastive pre-training in Table 11b. The baseline method follows EgoVLP [39] and uses a total batch size of 512. We observe that the performance improves when increasing the batch size to 1,024. The improvment diminishes if we further increase the batch size to 2,048. Therefore, we use 1,024 as the default batch size to get our main results in § 5.1.

Effect of projection dimension. We compare different choices of the projection head's dimension in Table 11c. We can see that using 256 achieves the best performance compared to 128 or 512.

Temperature in contrastive loss. In Table 12, we study the effect of different temperatures in the contrastive loss (Eq 4). Note that we switch to a batch size of 1,024 based on the observation in Table 11b. We start with a learnable temperature of 0.07 following CLIP [49]. We can see that

τ_r	learn	τ_n	learn	Avg. mAP	Avg. nDCG
0.07	✓	n/a	n/a	25.6	28.8
0.07	✓	0.07	✓	25.7	29.0
0.07	✓	0.10	✓	26.8	29.6
0.07	✓	0.10	✗	27.4	29.8
0.07	✗	n/a	n/a	26.0	29.0
0.07	✗	0.07	✗	**29.5**	**31.1**
0.07	✗	0.10	✗	27.4	29.8

Table 12. **Temperature in contrastive loss.** We observe that using a same fixed temperature for both NARRATOR's pairs and REPHRASER's pairs works better than all other settings.

using a higher initial temperature τ_n for the pairs generated by NARRATOR achieves noticable gain over the one that uses the same initial temperature of 0.07 for both τ_r and τ_n. We found that the within-batch accuracy during contrastive training for NARRATOR's pairs is significantly higher than the one for REPHRASER's pairs. Our conjection is that the dual-encoders is more likely to overfit the NARRATOR's pairs. Therefore, we switch to a fixed temperature and find that using $\tau_r = \tau_n = 0.07$ works better than all other settings, such as learnable temperature.

G. Qualitative Results

We provide more generated samples by our NARRATOR and REPHRASER in Figure 7. Note that our NARRATOR can generate reasonable captions from different views. For instance, Figure 7(d) illustrates that NARRATOR can describe the activities of both the camera wearer (starting with "C", which stands for "Camera wearer" in Ego4D) and the other person (starting with "O", which stands for "Observer" in Ego4D).

H. Licenses

The images in Figs. 2 to 4 and 7 are adapted from Ego4D videos. The video id (`$vid`) along with the start/end timestamp is provided below. The video can be viewed via the url `https://visualize.ego4d-data.org/$vid` (License is required for access).

- Figure 2: `1bfac46e-f957-4495-9583-dbd7fa683225, 01:30:00-01:50:00.`
- Figure 3 (top): `06919917-76bc-4adc-b944-2a722f165513, 00:00:08-00:00:10.`
- Figure 3 (bottom): `cf7c12db-1a9e-46d3-96d6-38174bbe373c, 00:21:17-00:21:19.`

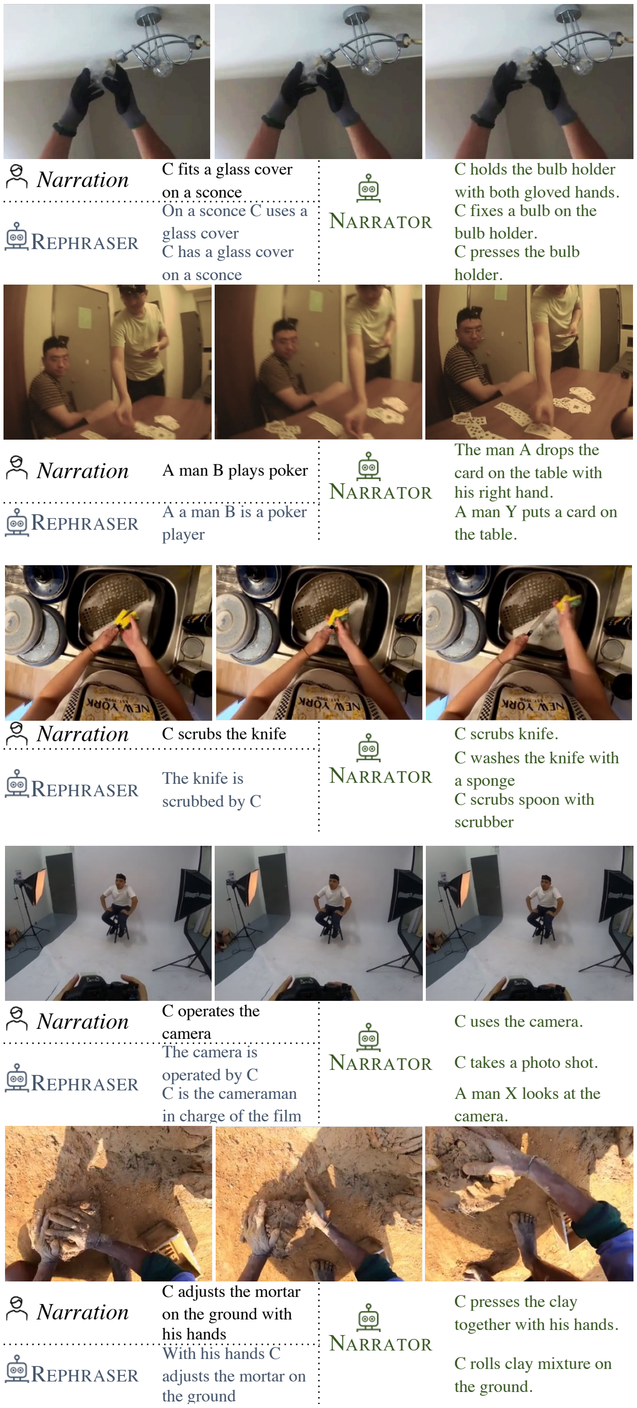

Narration
C fits a glass cover on a sconce
REPHRASER
On a sconce C uses a glass cover
C has a glass cover on a sconce
NARRATOR
C holds the bulb holder with both gloved hands.
C fixes a bulb on the bulb holder.
C presses the bulb holder.
Narration
A man B plays poker
REPHRASER
A a man B is a poker player
NARRATOR
The man A drops the card on the table with his right hand.
A man Y puts a card on the table.
Narration
C scrubs the knife
REPHRASER
The knife is scrubbed by C
NARRATOR
C scrubs knife.
C washes the knife with a sponge
C scrubs spoon with scrubber
Narration
C operates the camera
REPHRASER
The camera is operated by C
C is the cameraman in charge of the film
NARRATOR
C uses the camera.
C takes a photo shot.
A man X looks at the camera.
Narration
C adjusts the mortar on the ground with his hands
REPHRASER
With his hands C adjusts the mortar on the ground
NARRATOR
C presses the clay together with his hands.
C rolls clay mixture on the ground.

Figure 7. **More generated samples by our NARRATOR and REPHRASER on Ego4D**. NARRATOR generates new descriptions of the action taking place, potentially focusing on other objects or person being interacted with. REPHRASER not only changes the word order of the human narration but also diversifies it by using related verbs or nouns. Please refer to Appendix G for discussion.

- Figure 4: 3c0dffd0-e38e-4643-bc48-d513943dc20b, 00:00:12-00:00:14.
- Figure 7 (a): 26054ab4-4967-47b5-9b6c-e8a62f9295e0, 00:08:09-00:08:10.
- Figure 7 (b): 3130e00e-873a-4afb-93a6-7b07f3cf6597, 00:11:42-00:11:44.
- Figure 7 (c): def2e8dd-aaf7-467f-aa8f-46f654e6f4e0, 00:09:08-00:09:09.
- Figure 7 (d): ab865129-78fa-47d4-8a50-ff8c5533246f, 00:04:10-00:04:12.
- Figure 7 (e): 58a01f3a-52ce-4024-ab3c-b179caf4dafd, 00:28:43-00:28:45.