ACE: Action Concept Enhancement of Video-Language Models in Procedural Videos

Reza Ghoddoosian Nakul Agarwal Isht Dwivedi Behzad Darisuh
Honda Research Institute, USA
{reza_ghoddoosian, nakul_agarwal, idwivedi, bdariush}@honda-ri.com

Abstract

Vision-language models (VLMs) are capable of recognizing unseen actions. However, existing VLMs lack intrinsic understanding of procedural action concepts. Hence, they overfit to fixed labels and are not invariant to unseen action synonyms. To address this, we propose a simple fine-tuning technique, Action Concept Enhancement (ACE), to improve the robustness and concept understanding of VLMs in procedural action classification. ACE continually incorporates augmented action synonyms and negatives in an auxiliary classification loss by stochastically replacing fixed labels during training. This creates new combinations of action labels over the course of fine-tuning and prevents overfitting to fixed action representations. We show the enhanced concept understanding of our VLM, by visualizing the alignment of encoded embeddings of unseen action synonyms in the embedding space. Our experiments on the ATA, IKEA and GTEA datasets demonstrate the efficacy of ACE in domains of cooking and assembly leading to significant improvements in zero-shot action classification while maintaining competitive performance on seen actions.

1. Introduction

Understanding human actions in procedural videos—such as cooking or assembly—has numerous applications, including training, human-robot interaction, and anomaly detection. Accurate understanding of anomalies and proficiency is critical enabling targeted interventions, as they can compromise safety, efficiency, and overall effectiveness. Anomalies can appear as missed steps, redundant actions, deviations from sequences, or departures from expert performance [17, 19, 27]. Importantly, classification of previously unseen actions, as another form of anomaly, is essential for effective action recognition. For example, in a smart kitchen, it's impractical or unsafe to gather data for scenarios like "cutting finger" or "spilling hot water," yet an intelligent assistant must identify and respond to such actions accurately and in real time.

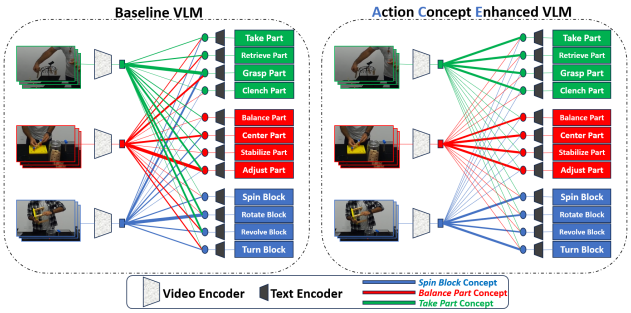

Figure 1. Illustration of the similarity between video and text representations for three action classes (concepts). Thicker lines indicate more similarity. Baseline VLMs (left) struggle with action synonym robustness. In contrast, ACE (right), improves accuracy in matching videos to action concepts, regardless of synonyms.

Vision-Language Models (VLMs) represent the state-of-the-art (SoTA) in zero-shot action recognition, where action categories are identified even if not explicitly seen during training. These models process videos and text through separate encoders, projecting them into a shared video-text embedding space. Here, a query video is matched to the closest text representation from unseen action labels manually curated by annotators. Since the actions and labels are unseen during training, VLMs must encode the broader concept of an action rather than the exact label. This enables the model to match a query video to its action class, regardless of the synonym used. Essentially, text representations describing the same action class should be projected close to each other in the embedding space. For example, as shown in Figure 1, a video of someone spinning a block should be associated with the relevant action class (denoted by the blue class), whether labeled "spin block," "rotate block," "revolve block," or "turn block."

Existing VLMs pretrained on large image-text datasets [49, 59, 60] often exhibit bias towards objects, failing to capture temporal action elements like verbs. Other VLMs [64, 65], pretrained on videos and internet transcripts, have text encoders that lack robustness, especially with fine-

grained action synonyms in specialized and procedural domains. To address this, we propose a fine-tuning technique called Action Concept Enhancement (ACE) to improve VLM robustness and concept understanding. To our knowledge, we are the first to investigate action concept understanding in VLMs' and test their classification robustness against fine-grained and unseen action synonyms.

We leverage the knowledge of a Large Language Model (LLM) to construct a synonym tree, where each node is an action label and its descendants are synonyms. During training, we use a classification loss function where videos are classified into novel combinations of action synonyms and their negatives, randomly chosen from the tree. This method generates numerous action label combinations, ensuring the model encounters new or rare action sets each iteration, simulating classification into unseen categories. The augmented synonyms introduce randomness and diversity, reducing overfitting to fixed verb representations, while negative labels help reduce bias toward objects. Our fine-tuning framework for VLMs integrates in-domain contextualization with the pretraining knowledge, enhancing recognition of unseen actions and understanding their concepts.

We evaluate our concept enhancement technique on the IKEA [5], ATA [19], and GTEA [16] datasets across cooking and assembly domains. Our method significantly outperforms the SoTA in recognizing unseen actions and understanding procedural concepts while performing competitively on seen actions. The contributions of this paper are:

- To the best of our knowledge, we are the first to evaluate the action concept understanding of VLMs by testing their robustness to procedural and unseen action synonyms.
- We introduce a fine-tuning mechanism that integrates in-domain knowledge into a pretrained model, enabling it to infer unseen procedural actions while maintaining performance on known actions.
- We use action synonyms stochastically during training to prevent VLMs from overfitting to fixed verbs and objects, leading to significant improvements in zero-shot action concept understanding.
- Our method, Action Concept Enhancement (ACE), is simple, generalizable, and easy to integrate into VLMs. We validate the classification efficacy of our method across different datasets and domains.

2. Related Work

Zero-Shot Action Recognition (ZSAR). ZSAR classifies videos into action categories not present in the training set. While transductive ZSAR uses test videos without labels during training, and generalized ZSAR handles both seen and unseen classes [14,32], we adopt an inductive approach, evaluating unseen and seen classes separately without access to unseen videos or labels during training. Earlier approaches used word embeddings [7, 42] or manually annotated class attributes [24] to represent action classes. Others decomposed text into fine-grained descriptions from the internet [10,46,57] and encoded them with BERT [11]. Methods like [29, 45] generated unseen visual prototypes from linear combinations of seen ones, [8] used reinforcement learning for video captioning to classification, and [25, 34] utilized object priors for action inference. Recently, VLMs [49, 54, 62], by aligning text and video embeddings through contrastive learning, have outperformed earlier methods, showing strong generalization to unseen classes. Our work builds on VLMs for recognizing both seen and unseen procedural actions.

Vision-Language Models for Action Recognition. VLMs have been applied to action understanding tasks like action localization [2, 13, 33, 40], alignment [21], and video retrieval [23, 28] in untrimmed videos. This paper focuses on step recognition in trimmed videos, where VLMs fall into two categories: image-based or video-based models. Image-based models, using a CLIP [47] encoder, leverage 400 million image-text pairs from the internet. These models adapt to video through prompt learning [26, 54], adding temporal layers [38, 41], or parameter-free fine-tuning on video datasets [30, 49, 60]. Despite large-scale pretraining, these methods struggle with knowledge retention and fail to capture temporal dynamics, focusing more on static objects than fine-grained action details like verbs. [38] addresses this by generating hard negative captions from the SMiT dataset [39], but improvements are mainly shown on Kinetics [9], a dataset known for static bias.

Video-based models pretrain temporal encoders, like TimeSformer [6], on large video datasets. Some models [4, 43] pretrain on Ego4D [20] for egocentric videos, while others [31, 35, 62, 64, 65] use Howto100M [36] instructional videos with auto-transcribed narrations. These models capture temporal action dynamics, but their text encoders, like CLIP [47], word2vec [37], MPNet [52], or BERT [11], struggle with synonym variability for unseen actions. We propose a mechanism to improve robustness to label synonym variations. Additionally, while most VLMs focus on cross-dataset zero-shot inference, we evaluate zero-shot action recognition in a base-to-novel setting, fine-tuning the encoders on seen actions and testing on unseen ones in the same dataset, as in [49].

Language Augmentation and Concept Learning. Augmenting language effectively distills more semantic knowledge into VLMs [12, 38, 44, 63]. In image understanding, methods like [18, 22, 55] add negative and hierarchical labels, while [50] uses stochastic captions to improve out-of-distribution image classification in CLIP models. More relevant to our work are methods that augment labels for video action understanding. Recent efforts [38, 56] focus on en-

hancing action knowledge by emphasizing verbs over objects. [56] uses tasks like video reversal and antonym detection to assess action knowledge, while [38] generates hard negatives by replacing verbs in captions. Our approach reduces overfitting to both objects and verbs, improving concept understanding in VLMs, and is tested on robustness to unseen verb synonyms.

Generating labels from LLMs has improved skeleton-based action understanding [61] and self-supervised action recognition [48]. [31] uses WikiHow step descriptions to align text and video during pretraining. BIKE [60] recently employed a frozen CLIP encoder to extract relative attributes from a lexicon as a similarity measure for category labels. [58] and [64] use LLMs to generate auxiliary captions for retrieval and video representation learning in untrimmed videos. In contrast, our approach integrates stochastic synonym augmentation during fine-tuning for unseen action recognition in trimmed videos. Before VLMs, synonyms were used in ZSAR by [3], but their transductive method included test videos during training.

3. Action Concept Enhancement (ACE)

3.1. Problem Definition

In training, our method processes a batch of size B from trimmed procedural videos $\{I_n\}_{n=1}^{B}$ and their ground-truth action indices $\{y_n\}_{n=1}^{B}$. y_n is the class index of the n^{th} video, corresponding to one of the C seen action categories $\boldsymbol{a} = \{a_i\}_{i=1}^{C}$. We define $\boldsymbol{a}$ as the *default* or *root* labels of seen action classes in the dataset. The goal is to fine-tune the pretrained vision encoder $\mathcal{E}()$ and text encoder $\mathcal{G}()$ so a trimmed test video is correctly classified into one of the action categories. This is achieved by aligning the query video embedding with the text embedding of its groundtruth action in the shared space.

We follow two separate classification scenarios at test time: first, classifying a test video into one of the seen classes $\boldsymbol{a}$; and second, classifying a test video into one of the previously unseen action labels $\acute{\boldsymbol{a}} = \{\acute{a}_i\}_{i=1}^{\acute{C}}$, regardless of the action synonyms used. This robustness is especially crucial for unseen actions, as the model has neither been optimized with nor expected any unseen action labels. Throughout this paper, bold notations distinguish sequences from single-element variables.

3.2. Action Verb Synonym Trees

We assume any procedural action a can be decomposed into a verb v and object[1] o pair, *i.e.*, $a = v \oplus o$. We also define $\mathcal{V}(a) \rightarrow v$ and $\mathcal{O}(a) \rightarrow o$ as functions that map action a to its corresponding verb and object components, respectively. Let $\boldsymbol{v}$ represent the set of $|\boldsymbol{v}|$ verb labels corresponding to root actions $\boldsymbol{a}$. As shown in Fig.2, for each $v_i \in \boldsymbol{v}$, we establish a tree structure where v_i is the root. In general, each parent node is a verb, and its M children nodes are its synonyms, along with the parent verb itself. Each parent node is replicated as a child to ensure previous information is preserved at every semantic level. Concretely, children of node v are denoted as $\boldsymbol{v}^{+} = \{(v^{+})_i\}_{i=1}^{M} = Synonyms(v) \cup \{v\}$, where $\cup$ is the union operation, and synonyms are generated by an LLM. Although the number of children remains consistent at each level within a tree, it can vary across different levels.

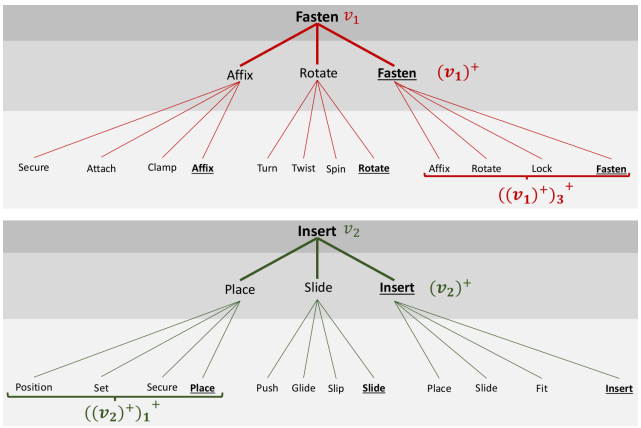

Figure 2. Synonym trees for the action verbs 'fasten' and 'insert' and sample notations. Each tree represents an action concept, with replicated parent nodes highlighted in bold. Some second-order synonyms provide broader descriptions of the action.

In this paper, we build each tree up to second-order synonyms (*i.e.*, synonyms of synonyms). However, in theory, these trees can extend to higher-order synonyms. Semantically, each tree corresponds to an action concept, and as trees deepen, action concepts overlap more and become less discriminative. This is because the connection between some higher-order synonyms and the root weakens, making action concepts coarser.

3.3. Stochastic Action Concept Learning

This section explains how the proposed synonym trees integrate into our learning framework.

Video-Label Alignment Loss. In line with VLM training, the video encoder $\mathcal{E}()$ and text encoder $\mathcal{G}()$ map the input video I_n and action labels $\boldsymbol{a}$ into a shared D-dimensional space. The cross-modal similarity $S(I_n, a_{y_n})$ between video I_n and its groundtruth label $a_{y_n} \in \boldsymbol{a}$ is maximized, while the similarity of I_n with other actions is minimized. The goal is to align related representations and separate unrelated ones. This alignment task is framed as a classification problem. For a batch of input data, the cross-entropy loss function L_{fixed} maximizes $P(n, \boldsymbol{a})$, the probability of I_n belonging to class a_{y_n} given action labels $\boldsymbol{a} = \{a_i\}_{i=1}^{C}$.

[1] Without loss of generality, the object component can include multiple objects and prepositions.

$$P(n, \boldsymbol{a}) = \frac{e^{S(I_n, a_{y_n})}}{\sum_{i=1}^{C} e^{S(I_n, a_i)}}, \tag{1}$$

$$L_{fixed} = -\frac{1}{B}\sum_{n=1}^{B} \log\big(P(n, \boldsymbol{a})\big). \tag{2}$$

Here, the cross-modal similarity $S(I, a)$ is defined as the average of cosine similarities between the video embedding and the text embeddings of M children of action a:

$$S(I, a) = \frac{1}{\tau M}\sum_{i=1}^{M} < \mathcal{E}(I), \mathcal{G}\big((a^{+})_i\big) >, \tag{3}$$

where τ is the pre-defined temperature, $< \cdot, \cdot >$ indicates cosine similarity between two normalized embeddings, and $(a^{+})_i = (\mathcal{V}(a)^{+})_i \oplus \mathcal{O}(a)$. Computing similarity with an action via the average of its synonyms has three main advantages: first, it brings related labels closer together through shared synonyms. Second, it helps to describe less familiar actions by their more recognizable synonyms. Third, it simply adds more in-domain textual data for the model to learn from.

Randomized Action Synonyms. We further model our action concept enhancement as an auxiliary classification task where the pool of available action labels is randomly augmented from the set of known root actions $\boldsymbol{a}$. Firstly, we define $\tilde{x}$ as a sample randomly selected from the set $\boldsymbol{x}$. Accordingly, $\widetilde{\mathcal{V}(a_i)^{+}}$ refers to a verb randomly sampled from the synonyms of verb $\mathcal{V}(a_i)$ associated with action a_i. Then, we leverage the verb synonym trees and Eq.1, to extend L_{fixed} by adding the auxiliary classification loss L_{rand} in Eq.5. Essentially, through L_{rand}, we categorize each video into one of the C action classes labeled by a new set of randomized action synonyms $\widetilde{\boldsymbol{a}^{+}}$ at each training iteration. In detail, as specified below, $\widetilde{\boldsymbol{a}^{+}}$ is a random augmentation of seen action classes, where each action class is represented by its randomly chosen verb synonym:

$$\widetilde{\boldsymbol{a}^{+}} = \big\{\widetilde{\mathcal{V}(a_i)^{+}} \oplus \mathcal{O}(a_i)\big\}_{i=1}^{C} = \{(\widetilde{a_i^{+}})\}_{i=1}^{C}, \tag{4}$$

$$L = -\frac{1}{B}\sum_{n=1}^{B} \log\big(P(n, \boldsymbol{a})\big) \underbrace{-\frac{1}{B}\sum_{n=1}^{B} \log\big(P(n, \widetilde{\boldsymbol{a}^{+}})\big)}_{L_{rand}}. \tag{5}$$

While for L_{rand} a new set of randomized action synonyms $\widetilde{\boldsymbol{a}^{+}}$ is constructed per training iteration, L_{fixed} uses the fixed root action labels throughout the entire training. Consequently, at every training iteration, each batch of videos is classified twice: once using the root labels and once using their randomized synonyms.

As root action labels are manually annotated in each dataset, they tend to be more precise descriptions of an action concept compared to AI-generated synonyms. Hence, the set of root labels in L_{fixed} is fixed and serves as a reference point. This enables the video-language encoders learn the connection between root action labels and their synonyms within each action concept subspace. "Concept subspace" refers to the space covering the text representations of all synonyms associated with an action in the joint space.

Meanwhile, variable action labels in L_{rand} prevent video-language encoders from overfitting to a single label, and instead learn different representations within a concept subspace. This enhances robustness to unseen action synonyms, and is beneficial in zero-shot recognition where actions and their labels are unknown. Our randomized augmentation technique can create up to M^C different action label combinations which are rarely repeated during training. Effectively, this simulates test time classification, where videos are categorized into unseen action labels.

Also, note that applying the similarity measure S to first order synonyms in Eq.5, allows VLMs to learn action concepts based on second order synonyms of the tree.

Shadow Negatives. We realized varying action synonyms through replacement of their verb components can bias the encoders to only objects. In other words, encoders learn to align videos to their correct action labels by only focusing on the object component, which defeats the purpose of concept learning. In order to alleviate this limitation, we introduce *shadow negative* as the $(C+1)^{th}$ category during classification. The shadow negative action shares the same object as the true action label; however, it pairs with a wrong verb. This approach compels the model to learn the verbs as well in order to accurately distinguish between the true label and its shadow negative. Specifically, we utilize the verb synonym trees to define the pool of shadow negative verbs $\boldsymbol{\mathcal{V}(a_i)^{-}}$ associated with the root action $a_i \in \boldsymbol{a}$ as:

$$\boldsymbol{\mathcal{V}(a_i)^{-}} = \bigcup_{j=1}^{C} \big(\boldsymbol{\mathcal{V}(a_j)^{+}} \setminus \boldsymbol{\mathcal{V}(a_i)^{+}}\big), \tag{6}$$

where '$\setminus$' refers to the set difference, *i.e.*, children of $\mathcal{V}(a_j)$ that are not among the children of $\mathcal{V}(a_i)$. At the beginning of each training iteration, for every class i, a shadow negative action $\widetilde{a_i^{-}}$ is constructed via random sampling from the pool of negative verbs $\boldsymbol{\mathcal{V}(a_i)^{-}}$ of that action:

$$\widetilde{a_i^{-}} = \widetilde{\mathcal{V}(a_i)^{-}} \oplus \mathcal{O}(a_i). \tag{7}$$

Then, we update $P(n, \boldsymbol{a}, \widetilde{a_{y_n}^{-}})$ as the probability of video I_n belonging to class $a_{y_n} \in \boldsymbol{a}$ given the pool of positive action labels $\boldsymbol{a}$ and shadow negative $\widetilde{a_{y_n}^{-}}$. Adding the shadow negative associated with the true action label of each video, extends the classification to $C+1$ classes:

$$P(n, \boldsymbol{a}, \widetilde{a_{y_n}^{-}}) = \frac{e^{S(I_n, a_{y_n})}}{\sum_{i=1}^{C} e^{S(I_n, a_i)} + e^{S(I_n, \widetilde{a_{y_n}^{-}})}}. \tag{8}$$

As a result, the final loss is also modified as follows:

$$L_f = -\frac{1}{B}\sum_{n=1}^{B}\Big(\underbrace{\log\big(P(n,\boldsymbol{a},\widetilde{a^-_{y_n}})\big)}_{L_{fixed}} + \underbrace{\log\big(P(n,\widetilde{\boldsymbol{a}^+},\widetilde{a^-_{y_n}})\big)}_{L_{rand}}\Big). \tag{9}$$

3.4. Training and Inference

Training: In Alg. 1, we summarize how to integrate ACE into the fine-tuning of VLMs for the task of video classification. We begin our algorithm by building the verb synonym trees $\{T(\mathcal{V}(a_i))\}_{i=1}^{C}$. Next, at the beginning of each training iteration, as we process a batch, new randomized sets of action synonyms $\widetilde{\boldsymbol{a}^+}$ and shadow negatives $\widetilde{\boldsymbol{a}^-}$ are generated. These, along with root labels $\boldsymbol{a}$ and their respective children are encoded by the text encoder. Through Eq. 9, our algorithm then engages each encoded video into two classification tasks involving $C+1$ categories. Consequently, this process encourages $\mathcal{E}$ and $\mathcal{G}$ encoders to explore action concepts by stochastically aligning videos and synonyms within their corresponding concept subspace.

Algorithm 1 Action Concept Enhancement (ACE)

Input: Input data $\mathcal{D}$ with videos and their groundtruth indices, the set of root action labels $\boldsymbol{a}$, and pretrained video and text encoders $\mathcal{E}$ and $\mathcal{G}$ respectively.

Output: Fine-tuned video encoder $\mathcal{E}$ and text encoder $\mathcal{G}$ with enhanced robustness to unseen action synonyms.

$\{T(\mathcal{V}(a_i))\}_{i=1}^{C} \leftarrow$ LLM($\boldsymbol{a}$) ▷ Verb synonym tree $T(\mathcal{V}(a_i))$ rooted in action verb $\mathcal{V}(a_i)$ for $a_i \in \boldsymbol{a}$.
for every epoch **do**:
  for batch $\{I_n, y_n\}_{n=1}^{B} \leftarrow \mathcal{D}$ **do**:
    $\widetilde{\boldsymbol{a}^+} = \{\widetilde{a_i^+} \leftarrow T(\mathcal{V}(a_i))\}_{i=1}^{C}$ ▷ Eq.4
    $\widetilde{\boldsymbol{a}^-} = \Big\{\widetilde{a_i^-} \leftarrow \{T(\mathcal{V}(a_j))\}_{j=1}^{C}\Big\}_{i=1}^{C}$ ▷ Eq.7
    $L_f \leftarrow$ Use Eq. 9 to calculate L_{fixed} and L_{rand}
    Backprop and optimize $\mathcal{E}$ and $\mathcal{G}$
  end for
end for
return Action Concept Enhanced (ACEd) $\mathcal{E}$ and $\mathcal{G}$

Inference: During inference, we classify query video I_n into the action class that has the highest similarity S with the query video, *i.e.*, $\text{argmax}_{a\in\mathbb{A}}\, S(I_n, a)$. Following [49], inference is done in two separate modes of *base* and *novel*, where $\mathbb{A}$ is the set of known classes $\boldsymbol{a}$ in the base mode and the set of unseen classes $\acute{\boldsymbol{a}}$ in the novel mode. In addition, in both base and novel modes, synonym trees are constructed, so $\mathbb{A}$ can be represented by the root action labels or the synonyms of the root labels. Note, we do not use any shadow negatives during inference.

4. Experiments

4.1. Experimental Setup

Datasets. We assess the efficacy of ACE in two procedural domains, cooking and assembly, using the following three *real-time* datasets: 1) *ATA* [19] is a toy assembly dataset with 12k trimmed videos and 15 action classes, recorded by 4 exocentric cameras. ATA's training and test splits consists of 27 and 4 participants, respectively. 2) *IKEA* [5] features table and drawer assemblies from 3 exocentric viewpoints, with 16k trimmed videos and 31 action classes. It's divided into 5 splits based on the environment, and results are averaged across these splits unless noted otherwise. 3) *GTEA* [16] is an egocentric dataset where 4 subjects prepare 7 different dishes. It comprises 525 videos and 10 action verb classes. Following [15], we perform classification on the verb classes in order to challenge zero-shot classification methods that are biased to objects and ignore verbs. We use 4 fold cross-validation with a subject left out for testing each time.

Evaluation Protocol. In each dataset, we select one-third of action classes with the least frequent verb as the set of unseen *(novel)* actions for zero-shot classification. The remaining classes are used as the seen *(base)* actions. This strategy increases the chance that novel and base verb sets are also mutually exclusive. The base actions in the training split are used for fine-tuning, while the base and novel actions in the test set are used to evaluate seen and unseen action recognition, respectively.

Evaluation Metrics. We report classification results using Top 1 Accuracy (*acc*) and *F1* score. While *acc* is averaged over all videos and is the most commonly used metric [49], it is skewed towards classes with more samples. In contrast, *F1* is computed as the average F1 score of all classes in the test split and weighs classes equally. Additionally, in order to test concept understanding and robustness to unseen action synonyms, we show the mean and standard deviation (std) over 10 different test runs. In each run, we repeat the classification of test videos, using a new combination of randomly-selected unseen action synonyms. Hence, we refer to these experiments as the Synonym Robustness Test (SRT). To ensure fairness, SRT for all methods is based on the same sets of action synonyms. We provide our generated sets of unseen action synonyms in the supp. material.

Implementation Detail. We use a 12-layer TimeSformer [6] video encoder pretrained on Howto100M via ProcVLR [65] and the original 12-layer CLIP text encoder (ViT-B/16). GPT-4 [1] generates synonyms for our method. The number of first and second order children in synonym trees are 2,9 and 11 for the IKEA, GTEA and ATA datasets, respectively. Furthermore, batch size is set to 16, temperature τ is adjusted to 0.02, and SGD optimizes the model for up to 15 epochs. Refer to the supp. material for more details.

Table 1. Procedural action classification on 2 exocentric datasets. HM is the Harmonic Mean of seen and unseen results [49]. SRT measures the robustness of VLMs to unseen action synonyms via mean±std. Results follow the '{*acc*}/{*F1*}' format with 2nd bests underlined.

		ATA Dataset *[10 base and 5 novel classes]*				IKEA Dataset *[21 base and 10 novel classes]*			
		Default Labels			SRT	Default Labels			SRT
Method	Pretraining Videos	Seen	Unseen	HM	Unseen	Seen	Unseen	HM	Unseen
Random	-	-	20.0/18.5	-	20.0±0/18.5±0	-	10.0/7.1	-	10.0±0/7.1±0
ViFi (ft) [49]	Just Images	93.1/94.2	33.5/25.4	49.3/40.0	44.8±5.4/33.2±3.7	78.7/61.0	44.3/29.6	56.7/39.9	34.2±7.5/27.6±4.6
ViFi (pr) [49]	ASM101 [51]	88.5/91.2	41.1/29.0	56.1/44.0	33.4±7.4/27.2±3.5	73.8/56.3	36.1/20.5	48.5/30.1	32.5±3.8/23.1±3.4
BIKE [60]	K400 [9]	93.0/92.2	29.8/18.8	45.1/31.2	25.6±11.7/17.7±6.9	77.2/63.7	36.5/31.3	52.2/43.3	27.3±10.9/30.1±7.3
Text4Vis [59]	K400 [9]	**93.5/95.0**	38.7/30.4	54.7/46.1	42.7±8.6/26.0±6.7	**87.6/71.6**	39.3/39.4	54.3/50.8	22.8±10.7/33.1±5.8
ProcVLR [65]	Howto100M [36]	89.8/90.1	37.1/31.3	52.5/46.4	43.8±10.6/33.0±6.9	82.3/62.5	37.0/30.3	51.0/40.8	32.1±13.6/27.1±5.5
Ours	Howto100M [36]	90.0/91.7	**60.8/47.3**	**72.6/62.4**	**59.4±3.2/43.3±4.2**	82.8/63.9	**54.5/45.5**	**65.7/53.2**	**45.9±6.1/41.1±4.2**

Table 2. Action classification of the egocentric GTEA videos.

	GTEA Dataset *[6 base and 4 novel classes]*			
	Default Labels			SRT
Method	Seen	Unseen	HM	Unseen
Random	-	25/22.8	-	25.0±0/22.8±0
ViFi (ft) [49]	75.9/72.8	27.5/21.1	40.4/32.7	33.1±15.5/20.8±14.0
ViFi (pr) [49]	58.4/50.2	25.6/13.1	35.6/20.8	24.4±20.7/15.0±14.9
BIKE [60]	63.8/62.5	50.0/33.9	56.1/ 43.9	39.9±30.6/29.7±12.7
Text4Vis [59]	82.2/81.3	63.8/**47.4**	71.8/**59.8**	40.0±27.4/30.3±22.3
ProcVLR [65]	68.0/64.2	50.3/36.1	57.8/46.2	45.0±16.9/28.8±15.5
Ours	**85.1/84.4**	**67.2**/41.0	**75.1**/55.2	**45.0±16.8/32.4±14.1**

Table 3. Impact of different modules of ACE on unseen actions.

	ATA Dataset		IKEA Dataset	
Setting	*acc*	*F1*	*acc*	*F1*
w/o leaf augmentation	45.2±11.6	33.1±8.1	34.0±9.9	33.1±6.4
w/o shadow negatives	56.4±2.9	36.8±3.0	35.4±9.3	36.3±4.0
w/o L_{rand}	41.1±7.6	29.5±5.5	44.4±10.6	36.2±4.2
w/o L_{fixed}	57.7±3.5	42.5±4.5	40.0±7.5	36.3±5.0
ACE	**59.4±3.2**	**43.3±4.2**	**45.9±6.1**	**41.1±4.3**

We intend to release our code and parameters publicly.

4.2. Comparison with Baselines

Baselines. 1) *Random* guess accuracy and F1 score, which is calculated considering the label distribution and equal guessing probability for each unseen action. This provides context to zero-shot prediction of other methods. 2) *ViFi (ft)* [49] fine-tunes non-frozen CLIP image and text encoders using averaged video frame encodings. 3) *ViFi (pr)* is a variation of [49], where image and text encoders are pretrained on procedural videos of the ASM101 dataset [9] and are kept frozen afterwards. Instead, learnable prompting layers are integrated in both encoders and trained during fine-tuning. 4) *BIKE* [60] represents methods that utilize text augmentation. It generates 200 extra language attributes by GPT-4 for training and inference. BIKE pretrains CLIP vision and text encoders separately on Kinetics-400 (K400) [9]. 5) *Text4Vis* [59] also uses CLIP encoders and pretrains on the K400 dataset while keeping the text encoder fixed as a classifier. 6) *ProcVLR* [65], similar to us, pretrains on Howto100M instructional videos and fine-tunes TimeSformer to encode videos. However, their CLIP text encoder is kept frozen.

We compare with these SoTA VLMs based on the availability of their codes and pretrained model checkpoints. Results are reported after running authors' source code on our procedural video datasets.

Procedural Action Classification. Table 1 and 2 compare our classification results with existing VLMs on exo and ego datasets, respectively. When tested against varying synonym labels, through SRT, our method shows the most robust performance. It achieves the highest mean *acc* and *F1* while maintaining low std in all datasets. GTEA test splits include fewer videos, which explains the higher overall std for all methods. While image-based models like ViFi also show low std for varying verbs, this is largely due to overfitting to objects, resulting in significantly lower mean values.

For the sake of completeness, we also compare results using default action labels. Our method significantly beats the SoTA in zero-shot classification for 5 out of 6 metrics across three datasets while remaining competitive on seen classes. Although Text4Vis [59] classifies seen actions more accurately in 2 of the 3 datasets, it doesn't generalize well to unseen actions, leading to lower overall performance (HM) compared to us. [59]'s success with base classes is mainly due to fine-tuning all encoder layers, whereas we only fine-tune the last three. However, as shown in Fig.4, fine-tuning more layers can further improve our seen action classification too. In general, the HM score reflects the trade-off between base and novel classes, and ACE improves the HM *acc* of the second best baselines by up to 16%, 11% and 4% for ATA, IKEA and GTEA datasets, respectively. Importantly, ProcVLR [65] is our one-to-one competitor as we share the same backbone and pretraining dataset. ACE enhances the action concept understanding of [65] consistently on all datasets for both base and novel actions.

4.3. Ablation Study and Fine-Tuning Analysis

We ablate our zero-shot Action Concept Enhancement based on the statistics over the 10 experiments of the Synonym Robustness Test. Results shown as '{*acc*}/{*F1*}'.

Table 4. Comparison of ACE when action synonyms are generated by GPT-4 vs. a human annotator on the GTEA dataset.

M=9	GPT Test Tree	Manual Test Tree
GPT Training Tree	45.0±16.8/32.4±14.1	56.3±13.9/44.3±17.7
Manual Training Tree	41.1±16.6/28.2±13.4	**63.6±16.4/45.4±15.3**

Ablation of ACE Components. Table 3 highlights how the zero-shot performance of ACE drops whenever each component of the algorithm is removed. We use the term "leaf augmentation" to describe representation of an action by the average of its synonym in the similarity measure of Eq.3. Leaf augmentation has the most significant impact on robustness, as it greatly reduces standard deviation. Also, Excluding shadow negatives leads to overfitting to objects, which reduces result variability but at the cost of lower overall average performance. This drop is less pronounced in the ATA dataset where unseen classes are easier to identify through objects alone. In contrast, the IKEA dataset involves multiple unseen actions that share the same objects, making action verb comprehension more crucial. Addition of L_{rand} also shows that the performance gain is not only due to extra text data in leaf augmentation, and the stochastic selection of synonyms plays a key role to prevent overfitting to fixed action verbs.

Sensitivity to the Quality of Action Synonyms. In order to evaluate how important the quality of generated synonyms are, an expert human annotator rebuilds the synonym trees by manually modifying the GPT-generated ones. The resulting manual synonyms fit the context of cooking better and there is no shared first order synonyms across different concepts. Manual synonym trees are built for both base and novel root actions in training and inference, respectively. We chose the GTEA dataset for this experiment (Table 4) as its scale makes the manual annotation plausible. As expected, better quality of manually-annotated synonyms improves the robustness of the model with the best result achieved when the model is trained and tested on manually-annotated synonym trees. Note, the mismatch between the manually-annotated synonyms during training and the GPT-generated synonyms during inference can negatively affect the model. Despite this, ACE still outperforms SoTA in zero-shot classification with GPT-generated synonyms. During SRT, the same randomization seeds are used for both trained models.

Sensitivity to the Number of Action Synonyms. Fig. 3 illustrates ACE's sensitivity to the number of first and second order children in the synonym trees. Monotonically non-increasing results on the ATA dataset show that more synonyms does not necessarily lead to better results. This is because the quality of synonyms is also a deciding factor (Table. 4), especially when the number of synonyms is small. However, evidently, using any number of added first or second order synonyms improves the VLM robustness compared to when no synonym is augmented. This impact is less significant on seen actions. Eventually, results tend to converge as the number of synonyms increases.

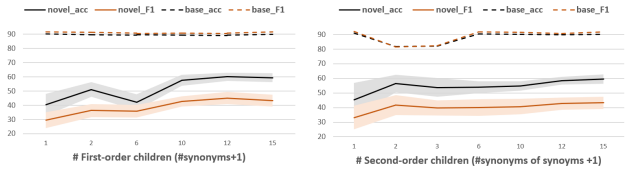

Figure 3. Impact of the quantity of augmented synonyms on mean and std (shaded area) for novel and base actions of the ATA dataset.

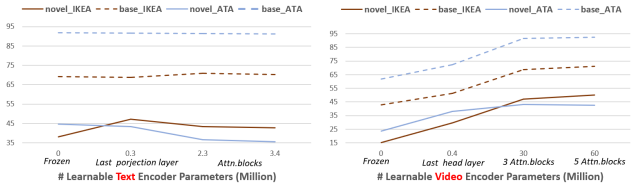

Figure 4. Impact of fine-tuning various layers of video-text encoders on the mean F1 score. Results on ATA and split 1 of IKEA.

How Many Encoder Layers to Fine-Tune? We study the extent to which each encoder should be fine-tuned in order to adjust the pretraining knowledge to new action concepts without losing prior information. Specifically, F1 scores in Fig.4-left show that deep finetuning of text encoder has minimal effect on the base actions, but degrades the zero-shot performance as the text encoder starts forgetting some of its pre-acquired knowledge. On the other hand, the video encoder (TimeSformer) can benefit from deeper fine-tuning (Fig.4-right). Not only this boosts results on seen actions, but also sometimes leads to better zero-shot performance as observed for the IKEA dataset. Nevertheless, fine-tuning more layers is computationally expensive and zero-shot performance gain is not always guaranteed on less diverse datasets such as ATA. Hence, we found the best trade-off to be training only the last three attention blocks of TimeSformer and the final projection layer of the text encoder.

Pretrained vs. In-Domain Knowledge. Table 5 shows that ACE effectively adjusts the prior knowledge of the pretrained TimeSformer by incorporating new in-domain information. Otherwise, applying ACE to a randomly-initialized TimeSfomer is not sufficient to learn action embeddings on our datasets. Moreover, although the pretrained TimeSformer excels in zero-shot performance on YouTube-based datasets like COIN [53, 65], it performs poorly on our real-time and unedited benchmarks without further fine-tuning. This is due to a domain shift, as TimeSformer was trained on YouTube procedural videos.

4.4. Concept Space Visualization

Fig. 5 shows the TSNE visualization of action concept subspaces in the IKEA dataset. Each action subspace is

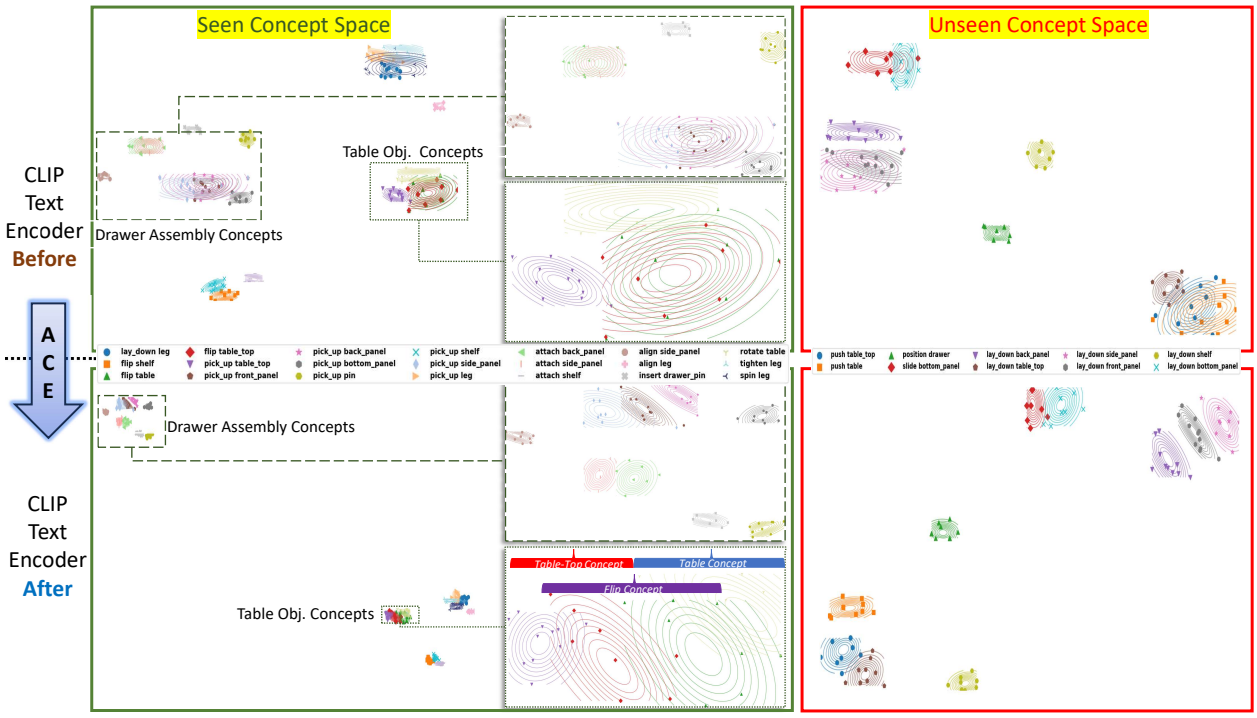

Figure 5. TSNE visualization for synonym embeddings in seen (outlined in green) and unseen (outlined in red) action spaces of IKEA dataset. The original CLIP embeddings of action synonyms are ACEd and grouped more distinctly. Please zoom in to see finer details.

Table 5. Impact of pretraining the TimeSformer on Howto100M before ACE fine-tuning. Random initialization used otherwise.

Setting		ATA		IKEA-split1	
Pretraining	ACE	Seen	Unseen	Seen	Unseen
✓	×	-	18.8/10.8	-	6.0/5.0
×	✓	24.6/9.7	20.5/13.2	21.9/1.7	34.0/8.0
✓	✓	81.8/81.5	56.4/41.6	88.0/71.1	58.6/50.2

estimated by a Gaussian resembling a galaxy, containing its synonym embeddings (planets) encoded by the text encoder. The top and bottom halves of Fig.5 compare synonym embeddings from the original CLIP (trained on 400M web image-text pairs) and ACEd CLIP for both base and novel action classes. For base classes, only two representative synonyms were seen during training, with the rest unknown to ACEd CLIP. Zoomed-in views of synonym embeddings for drawer and table-top classes are provided.

When comparing ACEd CLIP with the original CLIP in both seen and unseen spaces, ACEd embeddings have a smaller standard deviation, with action synonyms aligned more closely. This increases distinction between subspaces of different action classes and makes the model more invariant to synonyms. For example, while *push table*", *push table-top*", and *lay down table-top*" overlap in original CLIP, their ACEd embeddings are grouped more distinctly. This is notable since none of these concepts or their synonyms were seen during training. Additionally, ACEd CLIP better aligns similar concepts, unlike original CLIP, which lacks hierarchical understanding. In the seen concept space, embeddings for the drawer and table assemblies are close, while ACEd embeddings for different furniture assemblies are mapped far apart. Similarly, in the unseen concept space, synonyms of *lay down shelf*" initially project between drawer and table concepts but shift toward table assemblies after ACE, even though "shelf" is not used in other unseen action classes.

5. Conclusion

This paper focused on recognizing unseen procedural steps in trimmed videos. We demonstrated that current vision-language models lack intrinsic action concept understanding and tend to overfit to fixed labels. To address this, we introduced Action Concept Enhancement (ACE), a fine-tuning technique that improves VLMs' robustness and conceptual understanding. By integrating action synonyms stochastically during training, ACE mitigates overfitting to fixed verbs and objects. Our experiments in the cooking and assembly domains show that ACE significantly enhances zero-shot action concept recognition while maintaining competitive performance on seen actions.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 5

[2] Triantafyllos Afouras, Effrosyni Mavroudi, Tushar Nagarajan, Huiyu Wang, and Lorenzo Torresani. Ht-step: Aligning instructional articles with how-to videos. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[3] Ioannis Alexiou, Tao Xiang, and Shaogang Gong. Exploring synonyms as context in zero-shot action recognition. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 4190–4194. IEEE, 2016. 3

[4] Kumar Ashutosh, Rohit Girdhar, Lorenzo Torresani, and Kristen Grauman. Hiervl: Learning hierarchical video-language embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23066–23078, 2023. 2

[5] Yizhak Ben-Shabat, Xin Yu, Fatemeh Saleh, Dylan Campbell, Cristian Rodriguez-Opazo, Hongdong Li, and Stephen Gould. The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 847–859, 2021. 2, 5

[6] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 2, 5

[7] Biagio Brattoli, Joseph Tighe, Fedor Zhdanov, Pietro Perona, and Krzysztof Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4613–4623, 2020. 2

[8] Adrian Bulat, Enrique Sanchez, Brais Martinez, and Georgios Tzimiropoulos. Regen: A good generative zero-shot video classifier should be rewarded. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13523–13533, 2023. 2

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 2, 6

[10] Shizhe Chen and Dong Huang. Elaborative rehearsal for zero-shot action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13638–13647, 2021. 2

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[12] Sivan Doveh, Assaf Arbelle, Sivan Harary, Rameswar Panda, Roei Herzig, Eli Schwartz, Donghyun Kim, Raja Giryes, Rogério Schmidt Feris, Shimon Ullman, et al. Teaching structured vision & language concepts to vision & language models. 2023 ieee. In *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2657–2668, 2022. 2

[13] Nikita Dvornik, Isma Hadji, Ran Zhang, Konstantinos G Derpanis, Richard P Wildes, and Allan D Jepson. Stepformer: Self-supervised step discovery and localization in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18952–18961, 2023. 2

[14] Valter Estevam, Helio Pedrini, and David Menotti. Zero-shot action recognition in videos: A survey. *Neurocomputing*, 439:159–175, 2021. 2

[15] Yazan Abu Farha and Jurgen Gall. Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3575–3584, 2019. 5

[16] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, pages 3281–3288. IEEE, 2011. 2, 5

[17] Alessandro Flaborea, Guido Maria D'Amely di Melendugno, Leonardo Plini, Luca Scofano, Edoardo De Matteis, Antonino Furnari, Giovanni Maria Farinella, and Fabio Galasso. Prego: online mistake detection in procedural egocentric videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18483–18492, 2024. 1

[18] Yunhao Ge, Jie Ren, Andrew Gallagher, Yuxiao Wang, Ming-Hsuan Yang, Hartwig Adam, Laurent Itti, Balaji Lakshminarayanan, and Jiaping Zhao. Improving zero-shot generalization and robustness of multi-modal models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11093–11101, 2023. 2

[19] Reza Ghoddoosian, Isht Dwivedi, Nakul Agarwal, and Behzad Dariush. Weakly-supervised action segmentation and unseen error detection in anomalous instructional videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10128–10138, 2023. 1, 2, 5

[20] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 2

[21] Tengda Han, Weidi Xie, and Andrew Zisserman. Temporal alignment networks for long-term video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2906–2916, 2022. 2

[22] Lisa Anne Hendricks and Aida Nematzadeh. Probing image-language transformers for verb understanding. *arXiv preprint arXiv:2106.09141*, 2021. 2

[23] Jingjia Huang, Yinan Li, Jiashi Feng, Xinglong Wu, Xiaoshuai Sun, and Rongrong Ji. Clover: Towards a unified video-language alignment and fusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14856–14866, 2023. 2

[24] Dat Huynh and Ehsan Elhamifar. Fine-grained generalized zero-shot learning via dense attribute-based attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4483–4493, 2020. 2

[25] Mihir Jain, Jan C Van Gemert, Thomas Mensink, and Cees GM Snoek. Objects2action: Classifying and localiz-

ing actions without any video example. In *Proceedings of the IEEE international conference on computer vision*, pages 4588–4596, 2015. 2

[26] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision*, pages 105–124. Springer, 2022. 2

[27] Shih-Po Lee, Zijia Lu, Zekun Zhang, Minh Hoai, and Ehsan Elhamifar. Error detection in egocentric procedural task videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18655–18666, 2024. 1

[28] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4953–4963, 2022. 2

[29] Chung-Ching Lin, Kevin Lin, Lijuan Wang, Zicheng Liu, and Linjie Li. Cross-modal representation learning for zero-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19978–19988, 2022. 2

[30] Wei Lin, Leonid Karlinsky, Nina Shvetsova, Horst Possegger, Mateusz Kozinski, Rameswar Panda, Rogerio Feris, Hilde Kuehne, and Horst Bischof. Match, expand and improve: Unsupervised finetuning for zero-shot action recognition with language knowledge. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2851–2862, 2023. 2

[31] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13853–13863, 2022. 2, 3

[32] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9985–9993, 2019. 2

[33] Effrosyni Mavroudi, Triantafyllos Afouras, and Lorenzo Torresani. Learning to ground instructional articles in videos through narrations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15201–15213, 2023. 2

[34] Pascal Mettes, William Thong, and Cees GM Snoek. Object priors for classifying and localizing unseen actions. *International Journal of Computer Vision*, 129(6):1954–1971, 2021. 2

[35] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9879–9889, 2020. 2

[36] 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, 2019. 2, 6

[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[38] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15579–15591, 2023. 2, 3

[39] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14871–14881, 2021. 2

[40] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Zero-shot temporal action detection via vision-language prompting. In *European Conference on Computer Vision*, pages 681–697. Springer, 2022. 2

[41] 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 *European Conference on Computer Vision*, pages 1–18. Springer, 2022. 2

[42] AJ Piergiovanni and Michael Ryoo. Learning multimodal representations for unseen activities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 517–526, 2020. 2

[43] Shraman Pramanick, Yale Song, Sayan Nag, Kevin Qinghong Lin, Hardik Shah, Mike Zheng Shou, Rama Chellappa, and Pengchuan Zhang. Egovlpv2: Egocentric video-language pre-training with fusion in the backbone. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5285–5297, 2023. 2

[44] Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701, 2023. 2

[45] Shi Pu, Kaili Zhao, and Mao Zheng. Alignment-uniformity aware representation learning for zero-shot video classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19968–19977, 2022. 2

[46] Yijun Qian, Lijun Yu, Wenhe Liu, and Alexander G Hauptmann. Rethinking zero-shot action recognition: Learning from latent atomic actions. In *European Conference on Computer Vision*, pages 104–120. Springer, 2022. 2

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[48] Kanchana Ranasinghe and Michael S Ryoo. Language-based action concept spaces improve video self-supervised learn-

ing. *Advances in Neural Information Processing Systems*, 36:74980–74994, 2023. 3

[49] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6545–6554, 2023. 1, 2, 5, 6

[50] Shibani Santurkar, Yann Dubois, Rohan Taori, Percy Liang, and Tatsunori Hashimoto. Is a caption worth a thousand images? a controlled study for representation learning. *arXiv preprint arXiv:2207.07635*, 2022. 2

[51] Fadime Sener, Dibyadip Chatterjee, Daniel Shelepov, Kun He, Dipika Singhania, Robert Wang, and Angela Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21096–21106, 2022. 6

[52] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867, 2020. 2

[53] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 7

[54] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 2

[55] Weihan Wang, Zhen Yang, Bin Xu, Juanzi Li, and Yankui Sun. Vilta: Enhancing vision-language pre-training through textual augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3158–3169, 2023. 2

[56] Zhenhailong Wang, Ansel Blume, Sha Li, Genglin Liu, Jaemin Cho, Zineng Tang, Mohit Bansal, and Heng Ji. Paxion: Patching action knowledge in video-language foundation models. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3

[57] Shuang Wu, Sravanthi Bondugula, Florian Luisier, Xiaodan Zhuang, and Pradeep Natarajan. Zero-shot event detection using multi-modal fusion of weakly supervised concepts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2665–2672, 2014. 2

[58] Wenhao Wu, Haipeng Luo, Bo Fang, Jingdong Wang, and Wanli Ouyang. Cap4video: What can auxiliary captions do for text-video retrieval? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10704–10713, 2023. 3

[59] Wenhao Wu, Zhun Sun, and Wanli Ouyang. Revisiting classifier: Transferring vision-language models for video recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 2847–2855, 2023. 1, 6

[60] Wenhao Wu, Xiaohan Wang, Haipeng Luo, Jingdong Wang, Yi Yang, and Wanli Ouyang. Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6620–6630, 2023. 1, 2, 3, 6

[61] Wangmeng Xiang, Chao Li, Yuxuan Zhou, Biao Wang, and Lei Zhang. Generative action description prompts for skeleton-based action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10276–10285, 2023. 3

[62] 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. *arXiv preprint arXiv:2109.14084*, 2021. 2

[63] Zhongping Zhang, Yiwen Gu, Bryan A Plummer, Xin Miao, Jiayi Liu, and Huayan Wang. Movie genre classification by language augmentation and shot sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7275–7285, 2024. 2

[64] Yue Zhao, Ishan Misra, Philipp Krähenbühl, and Rohit Girdhar. Learning video representations from large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6586–6597, 2023. 1, 2, 3

[65] Yiwu Zhong, Licheng Yu, Yang Bai, Shangwen Li, Xueting Yan, and Yin Li. Learning procedure-aware video representation from instructional videos and their narrations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14825–14835, 2023. 1, 2, 5, 6, 7