MAtch, eXpand and Improve: Unsupervised Finetuning for Zero-Shot Action Recognition with Language Knowledge
Supplementary

Wei Lin[†1] Leonid Karlinsky[2] Nina Shvetsova[3] Horst Possegger[1]
Mateusz Kozinski[1] Rameswar Panda[2] Rogerio Feris[2] Hilde Kuehne[2,3]
Horst Bischof[1]

[1]Institute of Computer Graphics and Vision, Graz University of Technology, Austria
[2]MIT-IBM Watson AI Lab, USA
[3]Goethe University Frankfurt, Germany

For further insights into our approach MAXI, we introduce more dataset statistics (Sec. 1) and implementation details (Sec. 2) of MAXI.

In the additional results, we provide comparison of visualizations of attention heatmaps across several approaches in Sec. 3.1. Then we report results of few-shot action recognition (Sec. 3.2).

Furthermore, we perform ablation studies on (1) text bag filtering in Sec. 3.3 (2) finetuning with noisy action dictionary in Sec. 3.4, (3) different words to include in the text bag in Sec. 3.5, (4) different strategies to learn from words in a text bag in Sec. 3.6 and (5) text bag size in Sec 3.7.

We provide more examples of language sources used for training (Sec. 3.8). Lastly, we explore a cross-frame attention temporal module in Sec. 3.9.

1. Dataset Statistics

Kinetics-400 (K400) [10] is the most popular benchmark for action recognition tasks, containing around 240K training videos in 400 classes. The dataset consists of YouTube videos with an average length of 10 seconds. We use the training set of K400 for finetuning CLIP.

UCF101 [20] is collected from YouTube videos, consisting of 13K videos from 101 classes. There are three splits of training data (∼ 9.4K) and validation data (∼3.6K). Following XLCIP [16] and ViFi-CLIP [18], we report the average performance on the three validation splits.

HMDB51 [11] consists of 7K videos comprised of 51 action classes, collected from YouTube videos and movie clips. There are three splits of training data (∼ 3.5K, 70 videos per class) and validation data (∼1.5K, 30 videos per class). Following [16, 18], we report the average performance on the three validation splits.

Kinetics-600 (K600) [3] is an extension of K400, consisting of 650K videos in 600 classes. Following [5, 16, 18], we use the 220 classes[1] that are not included in K400 for zero-shot action recognition. There are three validation splits, each containing 160 classes randomly sampled from these 220 classes. We report the average performance on the three validation splits, each containing around 14K videos.

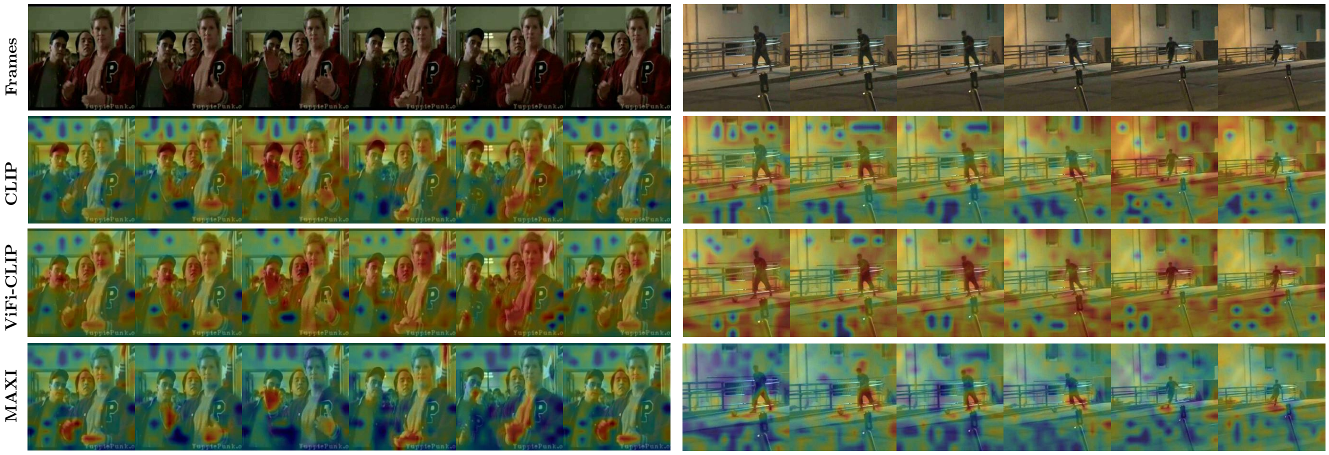
MiniSSv2 [4] (87 classes, 93K videos) is a subset of Something-Something v2 (SSv2) [7] (174 classes, 220K videos). SSv2 is an egocentric motion-based action dataset, which has a large visual domain shift to K400. Furthermore, the action classes are detailed descriptions of fine-grained movements, in a largely different language style than the K400 action dictionary, *e.g. Failing to put something into something because something does not fit*, and *Lifting a surface with something on it but not enough for it to slide down*. For zero-shot action recognition, we evaluate on the validation split of MiniSSv2 (12K videos). For few-shot action recognition, we follow [18] and evaluate on the validation split of SSv2 (25K videos).

Charades [19] is a long-range activity dataset recorded by people in their homes based on provided scripts for home activities. There are ∼10K videos in 157 classes. The average video length is 30 seconds. Each video has annotations of an average of 6.8 action instances, often in complex co-occurring cases. The validation split consists of 1.8K videos. We report the mean Average Precision (mAP) for the multi-label classification task.

Moments-in-Time (MiT) [15] is a large-scale action

† Correspondence: wei.lin@icg.tugraz.at

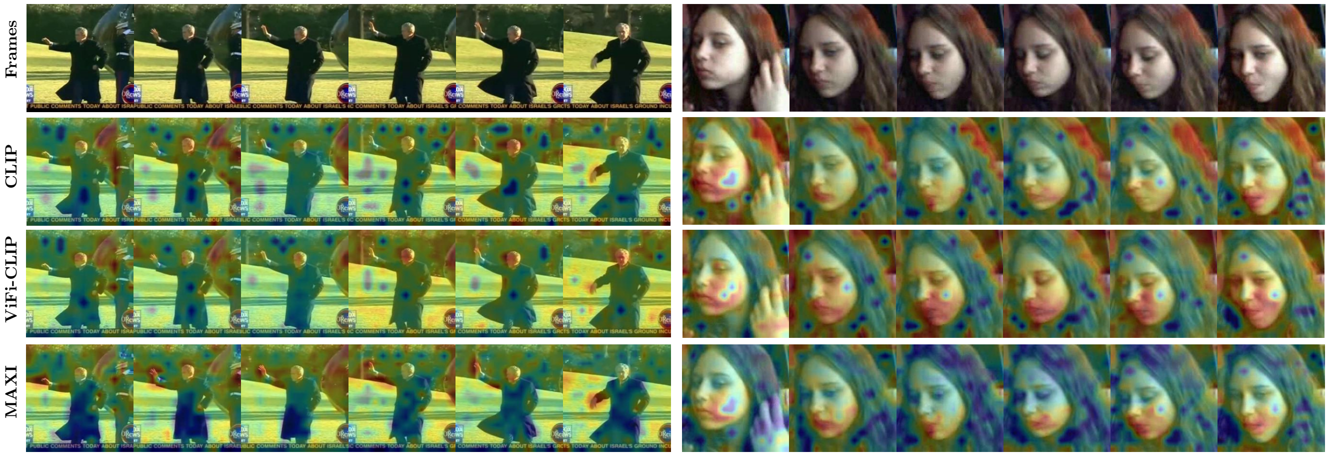
[1] In the evolution from K400 to K600, there are renamed, removed and split classes. See details in Appendix B in [5].



(a) clap

(b) kick ball

Figure 1. Attention heatmaps on actions which have a verb form (lemma or gerund) directly included in the K400 dictionary. We compare among CLIP (2nd row), ViFi-CLIP (3rd row) and our MAXI (4th row). Warm and cold colors indicate high and low attention. MAXI has more focused attention on hands (for *clap*) and legs (for *kick ball*).



(a) wave

(b) chew

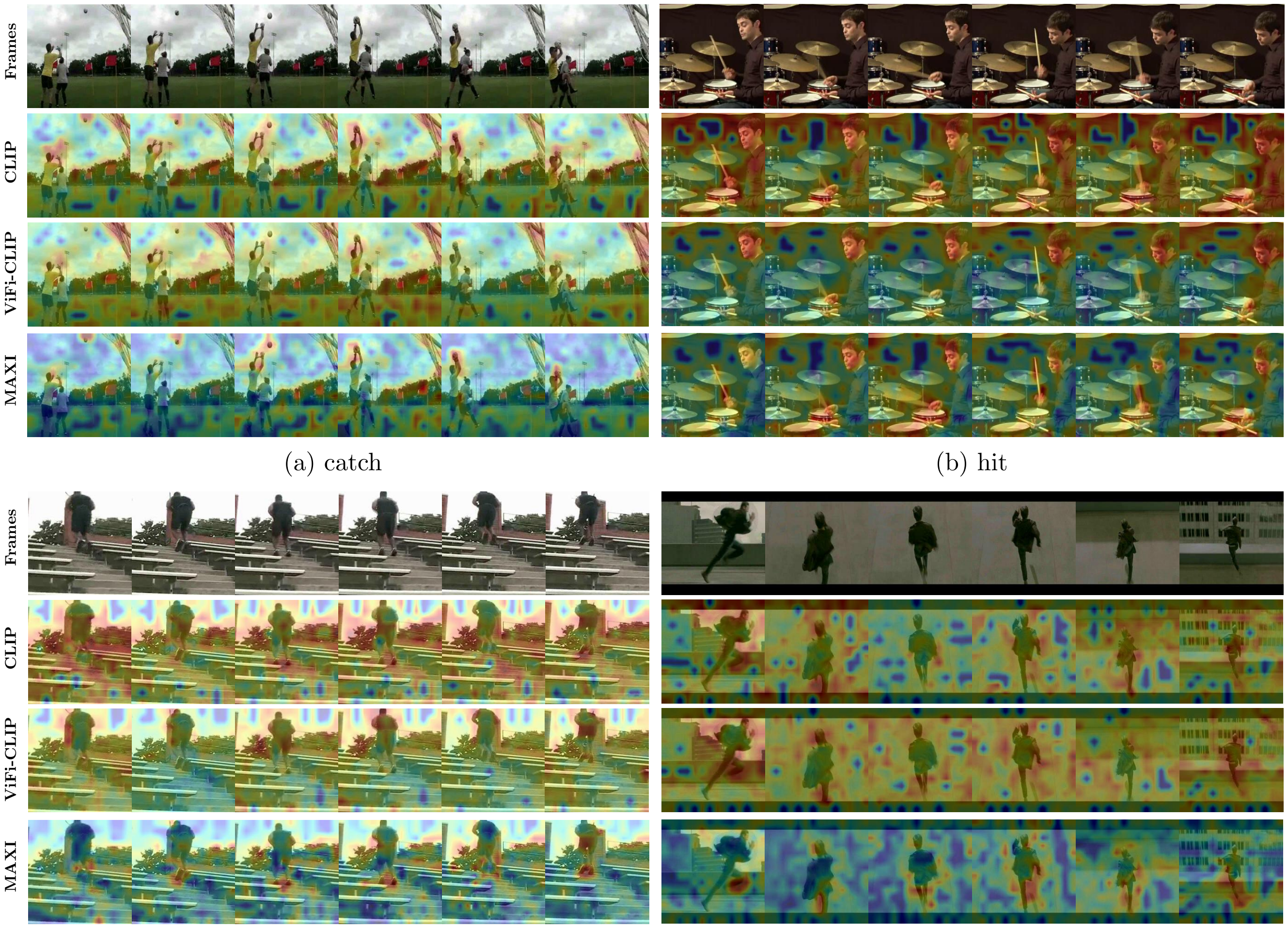
Figure 2. Attention heatmaps on novel actions which do not have any verb form included in the K400 dictionary. We compare among CLIP (2nd row), ViFi-CLIP (3rd row) and our MAXI (4th row). Warm and cold colors indicate high and low attention. MAXI has more focused attention on hand and arm for *wave*, and on the area of mouth for *chew*.

dataset of 3-second YouTube video clips, which cover actions in 305 classes, performed by both humans and animals. The validation split consists of 30K videos.

UAV Human (UAV) [13] is an action dataset recorded with an Unmanned Aerial Vehicle in unique camera viewpoints. There are 155 action classes. Actions in different categories are performed by a fixed group of subjects in the same background scenes. This leads to an extremely low object-scene bias and a large shift to the domain of K400 and CLIP. We evaluate on the RGB videos and report the average performance on the two official validation splits, each consisting of $\sim$ 6.2K videos.

2. Implementation Details

We employ CLIP with the ViT-B/16 [6] visual encoder. We follow the full-finetuning configuration of [18] to finetune both the visual and text encoder. We consistently set the temperature σ to 0.02. For zero-shot setting, we finetune on K400 without any ground truth labels. We use the AdamW optimizer [14] with an initial learning rate of 5×10^{-6} and a cosine decay scheduler. We sample 16 frames from each video and train with a batch size of 256 for 10 epochs. For few-shot learning, we sample 32 frames per video. We set the learning rate to 2×10^{-6}, and train with a batch size of 64 for 50 epochs. During inference,



(a) catch

(b) hit

(c) jump

(d) run

Figure 3. Attention heatmaps on actions which have a verb form (lemma or gerund) directly included in the K400 dictionary. We compare among CLIP (2nd row), ViFi-CLIP (3rd row) and our MAXI (4th row). Warm and cold colors indicate high and low attention. MAXI has more concentrated attention on the part where the action happens, *e.g.* catching ball with hands (Fig. 3(a), 4th row), hitting drum with stick (Fig. 3(b), 4th row), legs and feet jump on stairs (Fig. 3(c), 4th row), and attention on the running body (Fig. 3(d), 4th row).

we sample 1 view from each video. Inspired by [8, 23], we perform linear weight-space ensembling between the original CLIP (with ratio of 0.2) and the finetuned model. In the main results, we set the text bag filtering ratio p to 90% and bag size to 16.

We follow the evaluation protocol of zero-shot and few-shot action recognition from [16, 18]. We report mAP for multi-label classification on Charades and Top1/Top5 accuracy for single-label classification on the remaining datasets.

CLIP matching. The CLIP matching step is for consuming the language source of the predefined action dictionary D. We use CLIP[2] [17] with the ViT-B/16 visual encoder [6] to match each video with texts in the predefined action dictionary. To improve the matching quality for Text Bag Construction, we perform prompt ensembling over the 28 prompt templates[3] which are proposed by CLIP for Kinetics videos. Important to note, during inference we follow the exact protocol of ViFi-CLIP [18] and use only a single prompt.

GPT-3 text expansion. We employ the GPT-3 `text-davinci-003` model [2]. We set the temperature to 0.4. We generate 5 verb phrases using the input instruction - *Generate 5 phrases to describe the action of <action> in simple words*. Here for a video x_i, *<action>* is the best matched text $\hat{t}_i$ from the predefined action dictionary.

BLIP captioning. We use BLIP model [4] [12] with ViT-

[2] CLIP model source

[3] `https://github.com/openai/CLIP/blob/main/data/prompts.md`

[4] BLIP model source

L/16 as the image captioner. For each video, the image captioning is performed on 8 uniformly sampled frames. The frames are resized into 384.

Text augmentation. We use the natural language processing tool spaCy[5] to parse the verbs and verb phrases from the descriptions. We perform augmentation by converting the verbs into forms of lemma and gerund (present participle) and include results in the text bag.

Training. We employ CLIP with the ViT-B/16 visual encoder. We follow the full-finetuning configuration of [18] to finetune both the visual and text encoder. During training, we follow the configuration of [16, 18] for visual augmentation of multi scale crop, random flipping, color jitering and gray scaling. We do not perform augmentations of MixUp or CutMix.

As different videos have varying numbers of texts in their bags, we randomly sample N_{bag} texts from the originally constructed bag in each training iteration. For multiple instance learning, we use all the N_{bag} words in a text bag to form N_{bag} text prompts for each video. The text prompt is in the format of <*text1*> + <*text2*>. The first part <*text1*> is uniform for all the N_{bag} text prompts. Specifically, we use a hand-crafted prompt template *a photo of* <*action*>, where <*action*> is the best-matched text $\hat{t}_i$ from the predefined action dictionary (see Eq. 1 in the main manuscript). <*text2*> is an individual text from the text bag. To avoid duplication, we do not use $\hat{t}_i$ as <*text2*>.

Inference. We follow [16, 18] and sample a single view via sparse temporal sampling and spatial center crop. The same single prompt template is used in inference.

3. Additional Results

3.1. Attention Heatmaps

To gain more insights into the performance improvement of MAXI, we compare the visualizations of attention heatmaps across several approaches in Fig. 1, Fig. 2 and Fig. 3. *CLIP* is the original CLIP [17] without any finetuning. *ViFi-CLIP* [18] finetunes CLIP via supervised classification on K400 with ground truth annotations. *MAXI* is our approach of unsupervised finetuning with language knowledge.

We obtain the attention maps by computing the cosine similarity between the patch token features from the visual encoder and the text feature from the text encoder. We visualize the attention maps in several action classes from the downstream datasets used for the zero-shot action recognition task. Based on the relationship between the zero-shot action class and the K400 action dictionary used for training, we categorize the visualizations into 3 groups: (1) In-dictionary action classes which have a verb form (lemma or gerund) directly included in the K400 action dictionary, *e.g.* *clap* and *kick ball* in Fig. 1; (2) Novel actions classes which do not have any verb form included in the K400 action dictionary, *e.g.* *wave* and *chew* in Fig. 2; (3) General actions whose verb form is a basic component of several actions in the K400 action dictionary, *e.g.* *catch*, *hit*, *jump* and *run* in Fig. 3.

In-dictionary action classes. In Fig. 1, we visualize two samples of the action *clap* and *kick ball*. *clap* has the same lemma as *clapping* in the K400 dictionary, while *kick ball* has related actions of *kicking field goal* and *kicking soccer ball* in the K400 dictionary. We see that CLIP has incorrectly high attention on object (Fig. 1(a), 2nd row) or background scene (Fig. 1(b), 2nd row). ViFi-CLIP has cluttered high attention on both the subjects and the background scenes. On the contrary, MAXI has more focused attention on the hands (for *clap*) and legs (for *kick ball*).

In our GPT-3 text bag of *clapping*, related words such as *clap*, *smacking hands*, *slapping palms* and *clapping hands* are included. This strengthens the association between the action *clap* and the body part of hands, and leads to more accurate attention. Furthermore, in BLIP caption verb text bags, the verb *clap* appears several times in frame captions of K400 videos of *clapping*, *giving or receiving award* and *applauding*. This further improves the understanding of *clap*. Similarly, in BLIP frame captions, *kick* is an even more basic verb with large amount of occurrences.

Novel action classes. In Fig. 2, we compare the attention maps for the novel verbs *wave* and *chew* that do not appear in the K400 action dictionary. We see that for *wave*, CLIP and ViFi-CLIP have attention on the background scene or on the head, while MAXI has correct attention on the hand and arm. For *chew*, CLIP has more attention on the hair and ViFi-CLIP has attention on a large area of the face. On the contrary, MAXI has consistent focused attentions on the area of the mouth where the action *chew* happens.

The verb *wave* appears in BLIP caption verb text bags of several K400 videos of *clapping*, *applauding*, *celebrating*. The verb *chew* appears in captions of K400 videos of *eating carrots*, *eating spaghetti*, *eating watermelon* and *baby waking up*. The additional language source improves the knowledge of actions that never appear in the K400 action dictionary.

General actions. In Fig. 3, we illustrate the attention maps for four general verbs *catch*, *hit*, *jump* and *run*. These verbs are basic components of several actions in the K400 dictionary, *e.g.* *catching fish*, *catching or throwing frisbee*, *hitting baseball*, *jumping into pool* and *running on treadmill*. In these samples, CLIP and ViFi-CLIP have cluttered attention on the background scene or objects. MAXI has more concentrated attention on the part where the action happens, *e.g.* catching ball with hands (Fig. 3(a), last row), hitting drum with stick (Fig. 3(b), last row), legs and feet jump on stairs (Fig. 3(c), last row), and attention on the running body

[5] spaCy https://spacy.io/

(Fig. 3(d), last row).

These verbs are very general and could have highly diverse instantiations. *E.g*. *hit* (drum) in Fig. 3(b) is not close to *hitting baseball* on K400. *jump* (on stairs) in Fig. 3(c) is not close to *jumping into pool* or *bungee jumping* on K400, even if they share the same verb. In our GPT-3 verb bag and BLIP caption verb bag, there is a large amount of these verb instances that facilitate the comprehensive understanding of these general verbs. This leads to better focus even in unusual complex scenes, *e.g*. jumping on stairs (Fig. 3(c)).

3.2. Few-Shot Action Recognition

We perform few-shot all-way action recognition to evaluate the model learning capacity in a low data regime. In this setting, we specifically verify whether our self-supervised finetuning on K400 provides a proper initialization for few-shot learning. We follow the few-shot configuration of ViFi-CLIP [18] and XCLIP [16], and use the same training samples in 2, 4, 8 and 16-shot experiments without additional language source for a fair comparison. We train with 32 frames per video. We use the best backbone of self-supervised finetuning (from Sec. 3 in the main manuscript) as the model initialization for few-shot training. In Table 1, we report few-shot results of MAXI on three datasets, and also the zero-shot performance of our initialization as a reference. We compare with related approaches that directly perform few-shot learning on CLIP. For a fair comparison, we include the result of few-shot training with a CLIP model that is pre-trained with ground truth labels in the ViFi-CLIP paradigm.

In Table 1, we see that few-shot learning using a MAXI-pretrained backbone leads to best performance in most settings, even outperforming the fully-supervised pretrained backbone of ViFi-CLIP. The performance gap is significant in the more challenging extremely limited data scenarios (*e.g*. 2-shot on HMDB and UCF). Pretraining with full supervision as an initialization might lead to degraded performance in the following few-shot learning (*e.g*. 8-shot on HMDB, 4-shot on UCF), while our self-supervised finetuned model mitigates this problem, indicating improved generalizability.

3.3. Text bag filtering

To improve the quality of text bags used in training, we set a threshold δ_p on the similarity score from CLIP matching, such that $p \times 100\%$ of videos with highest similarity scores remain after the thresholding (see Sec. 2.2 in the main manuscript). We perform CLIP matching between unlabeled K400 videos and the K400 action dictionary, and use the filtered videos and text bags for finetuning CLIP. In Table 2, we report the matching accuracy (after filtering), and zero-shot transfer performance of models finetuned with the filtered K400 videos and text bags. As a reference, we also report CLIP zero-shot performance, and the case of finetuning on 100% accurate video-textbag pairs using ground truth annotation, which leads to the best zero-shot transfer on most datasets.

In Table 2, we notice that the CLIP matching accuracy increases continuously with decreasing filtering ratio p. Setting $p = 90\%$ leads to consistent improvement of zero-shot transfer, in comparison to the case of $p = 100\%$ due to improved quality of matched texts. Setting $p = 50\%$ leads to partial improvement compared to $p = 100\%$. Further reducing p to 50% leads to performance degradation due to the limited amount of data. This indicates that selecting text bags that CLIP is confident about ensures improved finetuning for more effective zero-shot transfer. However, there is a trade-off between the quality of the filtered data and the amount of data used for training.

3.4. Robustness against noisy action dictionary

In a practical scenario, we have coarse prior knowledge of the potential action types in an unannotated video collection, which defines an action dictionary. However, such knowledge might be noisy. We explore the robustness of our finetuning pipeline against such a noisy action dictionary. We consider two cases of noisy action dictionaries: (1) an under-specified dictionary consisting of only half of the words of the original K400 action dictionary. Specifically, we use the 200 action names from MiniKinetics [4] (a 200-class subset of K400). (2) An over-specified dictionary by adding noisy verbs and verb phrases into the original K400 action dictionary. We parse verbs from the captions in the validation set of the WebVid2.5M dataset [1], and randomly sample verbs to add to the dictionary. We increase the ratio of noisy verbs, and add 400, 800 and 1200 verbs into the dictionary.

In Table 3, we report the zero-shot transfer performance of models finetuned with these noisy dictionaries. Here we set the text bag filtering $p = 50\%$ for improved text bag quality. We also report the results with the original K400 action dictionary as a reference. Apparently, using the clean original K400 action dictionary leads to the best zero-shot transfer on most of the downstream datasets. However, using noisy action dictionaries still leads to significant performance boost compared to the CLIP zero-shot results without finetuning. We see that even with extremely noisy dictionary where 50% to 75% of words do not match with the video data, our finetuning still results in a robust zero-shot transfer performance to unseen datasets. This indicates the robustness of our pipeline with different cases of noisy predefined dictionaries. The robustness is the consequence of the fact that we collect knowledge from multiple language sources and learn from them via Multiple Instance Learning. Note that the zero-shot transfer does not have consistent change in performance across the downstream datasets,

Dataset	pretrain on K400	sett.	HMDB51				UCF101				SSv2			
Shots			2	4	8	16	2	4	8	16	2	4	8	16
CLIP [17]	no	ZS	41.9	41.9	41.9	41.9	63.6	63.6	63.6	63.6	2.7	2.7	2.7	2.7
ActionCLIP [22]	no	FS	47.5	57.9	57.3	59.1	70.6	71.5	73.0	91.4	4.1	5.8	8.4	11.1
XCLIP [16]	no	FS	53.0	57.3	62.8	64.0	48.5	75.6	83.7	91.4	3.9	4.5	6.8	10.0
A5 [9]	no	FS	39.7	50.7	56.0	62.4	71.4	79.9	85.7	89.9	4.4	5.1	6.1	9.7
ViFi-CLIP [18]	no	FS	57.2	**62.7**	64.5	**66.8**	80.7	85.1	90.0	92.7	6.2	7.4	8.5	**12.4**
MAXI	yes w/o gt	ZS	49.2	49.2	49.2	49.2	77.8	77.8	77.8	77.8	4.8	4.8	4.8	4.8
ViFi-CLIP [18]	yes gt	FS	55.8	60.5	64.3	65.4	84.0	86.5	90.3	92.8	6.6	6.8	8.6	11.0
MAXI	yes w/o gt	FS	**58.0**	60.1	**65.0**	66.5	**86.8**	**89.3**	**92.4**	**93.5**	**7.1**	**8.4**	**9.3**	**12.4**

Table 1. Few-shot action recognition on HMDB, UCF and SSv2. We report few-shot learning results with and without pretraining on K400.

Matching	ratio p	matching acc. on K400	UCF101	HMDB51	K600	MiniSSv2	Charades	UAV Human	Moments-in-time
CLIP [17] (w/o finetune) Zero-Shot			69.93	38.02	63.48	3.96	19.80	1.79	20.11
gt	100%	100%	**82.39**	**52.68**	**73.39**	5.61	**25.31**	**4.47**	**23.79**
CLIP matching	100%	59.7%	77.88	51.09	71.24	5.46	23.52	2.53	22.44
CLIP matching	90%	64.3%	78.17	52.24	71.43	**6.37**	23.79	2.72	22.91
CLIP matching	50%	80.9%	78.18	50.35	70.78	5.74	23.89	3.06	22.41
CLIP matching	30%	89.5%	76.71	47.73	70.57	4.92	23.14	2.89	21.96

Table 2. Text bag filtering with different filtering ratio p. We report the CLIP matching accuracy (after filtering) on K400, and the zero-shot transfer performance of models finetuned with the filtered K400 videos and text bags.

Action dictionary	dictionary size	UCF101	HMDB51	K600	MiniSSv2	Charades	UAV Human	Moments-in-time
CLIP [17] (w/o finetune) Zero-Shot		69.93 / 92.7	38.02 / 66.34	63.48 / 86.80	3.96 / 14.42	19.80	1.79 / 7.05	20.11 / 40.81
K400	400	**78.18 / 96.03**	**50.35 / 77.10**	**70.78 / 92.17**	**5.74 / 17.70**	**23.89**	**3.06 / 9.46**	**22.41 / 45.83**
MiniKinetics	200	75.10 / 95.82	48.34 / 76.95	69.23 / 90.92	**6.50 / 18.76**	22.70	2.40 / 8.04	**22.50 / 46.01**
K400+WebVid2.5M	800	75.99 / 96.00	45.97 / 73.94	69.14 / 91.13	4.81 / 15.79	22.67	2.11 / 8.00	20.92 / 43.99
K400+WebVid2.5M	1200	75.72 / 96.02	45.51 / 73.97	69.36 / 91.11	4.21 / 15.15	22.35	2.39 / 7.98	21.29 / 44.33
K400+WebVid2.5M	1600	76.14 / 96.01	44.84 / 71.79	69.23 / 91.10	4.42 / 14.71	22.89	2.14 / 7.71	20.69 / 43.59

Table 3. Robustness of finetuning with noisy action dictionaries. We add noisy verbs parsed from the WebVid2.5M dataset into the original K400 action dictionary. We report the zero-shot transfer performance (mAP on Charades and Top1/Top5 accuracy on other datasets). We set the text bag filtering ratio $p = 50\%$ for improved text bag quality.

Text bag	UCF101	HMDB51	K600
K400 dict.	76.45	47.43	69.98
K400 dict. + BLIP object nouns	76.23	50.15	71.13
K400 dict. + BLIP verbs	76.94	50.92	**71.25**
K400 dict. + GPT3 verbs	76.98	50.46	71.24
K400 dict. + GPT3 verbs + BLIP verbs	**77.88**	**51.09**	71.24

Table 4. Combinations of words in text bags. We report the zero-shot transfer performance on UCF, HMDB and K600. For a thorough ablation, we set the text bag filtering ratio $p = 100\%$ to keep the full noisy text bag property.

as different datasets have different language domain shift to the action dictionary used for training.

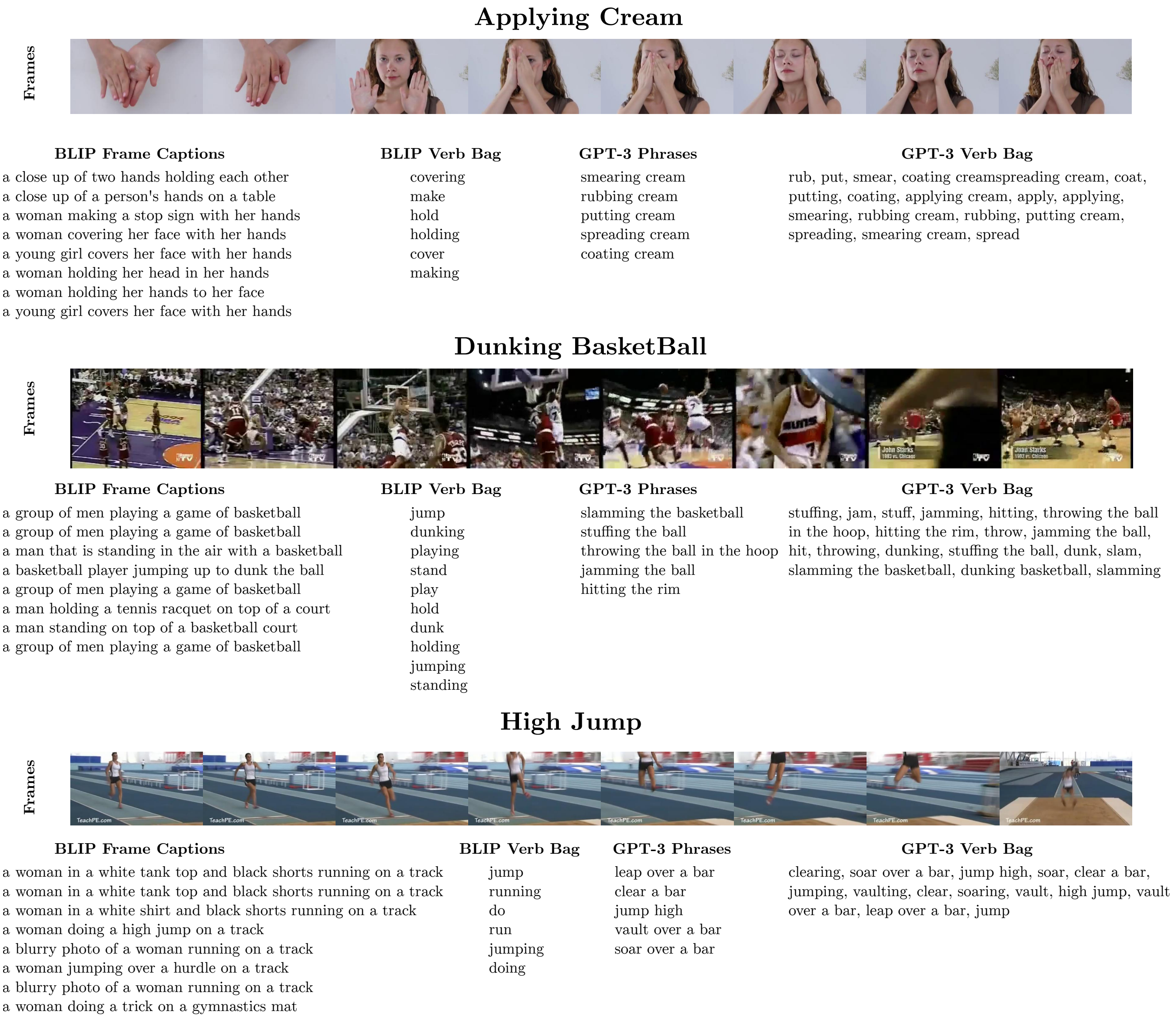
3.5. What words to include in the text bag?

In Table 4, we investigate different combinations of words to include in the text bag. Besides the original K400 action dictionary (*K400 dict.*), we explore: (1) *BLIP verbs*: verbs parsed from BLIP captions; (2) *BLIP object nouns*: nouns of objects parsed from BLIP captions; (3) *GPT3 verbs*: verbs and verb phrases from GPT3 text expansion. For a thorough ablation, we set the text bag filtering ratio $p = 100\%$ to keep the full noisy text bag property.

In Table 4, we notice that additional language source upon the original K400 action dictionary leads to further improvement in zero-shot transfer. Interestingly, using BLIP verbs has slightly better results than the case of BLIP object nouns. We assume this is because CLIP has a high object bias and is less sensitive to the language of verbs. Finetuning CLIP by injecting verbs leads to better zero-shot performance in action recognition. Consequently, combining BLIP verbs and GPT3 verbs in the text bag leads to the best zero-shot transfer.

3.6. How to learn from words in text bags?

In Table 5, we explore different strategies of learning from words in a text bag: (1) *Cross entropy*: classification in a fixed class space. (2) *NCE*: contrastive learning



BLIP Frame Captions

a close up of two hands holding each other
a close up of a person's hands on a table
a woman making a stop sign with her hands
a woman covering her face with her hands
a young girl covers her face with her hands
a woman holding her head in her hands
a woman holding her hands to her face
a young girl covers her face with her hands

BLIP Verb Bag

covering
make
hold
holding
cover
making

GPT-3 Phrases

smearing cream
rubbing cream
putting cream
spreading cream
coating cream

GPT-3 Verb Bag

rub, put, smear, coating creamspreading cream, coat, putting, coating, applying cream, apply, applying, smearing, rubbing cream, rubbing, putting cream, spreading, smearing cream, spread

BLIP Frame Captions

a group of men playing a game of basketball
a group of men playing a game of basketball
a man that is standing in the air with a basketball
a basketball player jumping up to dunk the ball
a group of men playing a game of basketball
a man holding a tennis racquet on top of a court
a man standing on top of a basketball court
a group of men playing a game of basketball

BLIP Verb Bag

jump
dunking
playing
stand
play
hold
dunk
holding
jumping
standing

GPT-3 Phrases

slamming the basketball
stuffing the ball
throwing the ball in the hoop
jamming the ball
hitting the rim

GPT-3 Verb Bag

stuffing, jam, stuff, jamming, hitting, throwing the ball in the hoop, hitting the rim, throw, jamming the ball, hit, throwing, dunking, stuffing the ball, dunk, slam, slamming the basketball, dunking basketball, slamming

BLIP Frame Captions

a woman in a white tank top and black shorts running on a track
a woman in a white tank top and black shorts running on a track
a woman in a white shirt and black shorts running on a track
a woman doing a high jump on a track
a blurry photo of a woman running on a track
a woman jumping over a hurdle on a track
a blurry photo of a woman running on a track
a woman doing a trick on a gymnastics mat

BLIP Verb Bag

jump
running
do
run
jumping
doing

GPT-3 Phrases

leap over a bar
clear a bar
jump high
vault over a bar
soar over a bar

GPT-3 Verb Bag

clearing, soar over a bar, jump high, soar, clear a bar, jumping, vaulting, clear, soaring, vault, high jump, vault over a bar, leap over a bar, jump

Figure 4. Examples of video frames, BLIP frame captions, GPT-3 phrases, together with the derived BLIP verb bag and GPT-3 verb bag. The videos are from the K400 dataset.

Objective	UCF101	HMDB51	K600
Cross entropy	74.48	48.69	65.09
NCE	77.26	49.85	70.08
MIL-Max	77.24	49.85	70.71
MIL-NCE only instance-level	76.96	50.48	70.14
MIL-NCE	**77.88**	**51.09**	**71.24**

Table 5. Different strategies of learning from text bags. We report the zero-shot transfer performance on UCF, HMDB and K600. For a thorough ablation, we set the text bag filtering ratio $p = 100\%$ to keep the full noisy text bag property.

Bag size	UCF101	HMDB51	K600
1	77.26	49.85	70.08
4	77.24	49.84	70.71
8	77.70	50.61	**71.35**
16	**77.88**	**51.09**	71.24

Table 6. Effect of bag size. We report the zero-shot transfer performance on UCF, HMDB and K600. For a thorough ablation, we set the text bag filtering ratio $p = 100\%$ to keep the full noisy text bag property.

to encourage instance-level match between a pair of video and text. In this case, we randomly sample one text from the text bag in each iteration. (3) *MIL-Max*: in each iteration, among words in a text bag, we choose the word with

Temp. attention layers	UCF101	HMDB51	K600	MiniSSv2	Charades	UAV Human	Moments-in-time
None	**78.17**	**52.24**	**71.43**	**6.37**	**23.79**	2.72	**22.91**
2	77.38	51.83	70.41	5.98	22.87	**2.90**	22.51
6	75.91	51.92	69.23	6.09	21.78	2.52	22.52

Table 7. Cross-frame temporal attention modules. we report the zero-shot transfer performance after finetuning CLIP on K400. We train with text bags of GPT3 verbs and BLIP verbs. We set the text bag filtering ratio $p = 90\%$. Adding temporal attention module does not lead to performance improvement.

the maximum similarity to the video, and pass the similarity in the contrastive loss. (4) *MIL-NCE*: as explained in Sec. 2.3 in the main manuscript, we softly associate a bag of texts with the video, and sum up the similarities of texts in a bag (5) *MIL-NCE only instance-level*: the *MIL-NCE* on instance-level match between video and text bag, without encouraging videos and text bags with the same best matched text to be close to each other (see Sec. 2.3 in the main manuscript). In Table 5, we see that cross entropy of classification in a fixed class space leads to the most inferior result, while our MIL-NCE achieves the best improvement. Encouraging videos and text bags with the same best matched text to be close to each other also leads to some performance boost in contrast to only instance-level matching.

3.7. Bag size

We perform an ablation on the bag size in Table 6. A bag size of 1 is the same as *NCE* loss with random word sampling in Table 5. Increasing the bag size from lower numbers to 8 leads to consistent performance improvements. Using bag size 16 has further slight performance boost. We report our main results with a bag size of 16.

3.8. Examples of Language Sources

Similar to the *cooking egg* example in Fig. 1 in the main manuscript, we illustrate more examples of video frames, BLIP frame captions, GPT-3 phrases, together with the derived BLIP verb bag and GPT-3 verb bag in Fig. 4. The videos are from the unlabeled K400 dataset which we use for training.

3.9. Parameter-Free Temporal Module

As mentioned in Sec. 2.1 in the main manuscript, we explore a parameter-free temporal-aware module on the CLIP model. We modify the multi head attention module [21] in the visual encoder of CLIP to be temporal aware. Originally, the attention on the frame t is computed via $A_t(Q_t, K_t, V_t) = \text{softmax}\frac{Q_t {K_t}^\top}{d_k} V_t$, where Q_t, K_t and V_t are the query, key and value from frame t.

We explore to compute the cross-frame attention via

$$A'_t(Q_t, K_{t+i}|_{i \in I}, V_t) = \text{softmax}\frac{\sum_{i \in I}(Q_t \cdot {K_{t+i}}^\top)/|I|}{d_k} V_t \tag{1}$$

where we set $I = \{-1, 0, 1\}$. In this case, we use the keys from the frame $t-1$, t and $t+1$ to compute the attention for frame t.

We apply the cross-frame attention on the last 2 and on the last 6 transformer layers in the visual encoder of CLIP. In Table 7, we report the zero-shot transfer performance. We see that in comparison to the variant without any temporal attention module, using cross-frame attention does not lead to performance improvement. K400 is of far smaller scale in comparison to the original CLIP domain. Finetuning from the CLIP model weights with a modified architecture could result in the case that the model drifts far away from the wise CLIP source domain. The results are consistent with the claims in [18] that a sophisticated temporal module does not necessarily lead to performance improvement.

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1728–1738, 2021. 5

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 3

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

[4] Chun-Fu Richard Chen, Rameswar Panda, Kandan Ramakrishnan, Rogerio Feris, John Cohn, Aude Oliva, and Quanfu Fan. Deep analysis of cnn-based spatio-temporal representations for action recognition. In *CVPR*, pages 6165–6175, 2021. 1, 5

[5] Shizhe Chen and Dong Huang. Elaborative rehearsal for zero-shot action recognition. In *ICCV*, pages 13638–13647, 2021. 1

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3

[7] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, pages 5842–5850, 2017. 1

[8] Gabriel Ilharco, Mitchell Wortsman, Samir Yitzhak Gadre, Shuran Song, Hannaneh Hajishirzi, Simon Kornblith, Ali Farhadi, and Ludwig Schmidt. Patching open-vocabulary models by interpolating weights. In *NeurIPS*, 2022. 3

[9] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, pages 105–124. Springer, 2022. 6

[10] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[11] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, pages 2556–2563. IEEE, 2011. 1

[12] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *arXiv preprint arXiv:2201.12086*, 2022. 3

[13] Tianjiao Li, Jun Liu, Wei Zhang, Yun Ni, Wenqian Wang, and Zhiheng Li. Uav-human: A large benchmark for human behavior understanding with unmanned aerial vehicles. In *CVPR*, pages 16266–16275, 2021. 2

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[15] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *TPAMI*, 42(2):502–508, 2019. 1

[16] 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*, pages 1–18. Springer, 2022. 1, 3, 4, 5, 6

[17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3, 4, 6

[18] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. *arXiv preprint arXiv:2212.03640*, 2022. 1, 2, 3, 4, 5, 6, 8

[19] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, pages 510–526. Springer, 2016. 1

[20] 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. 1

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017. 8

[22] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 6

[23] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *CVPR*, pages 7959–7971, 2022. 3