Open Vocabulary Multi-Label Video Classification

Rohit Gupta[1], Mamshad Nayeem Rizve[2], Jayakrishnan Unnikrishnan[2], Ashish Tawari[2], Son Tran[2], Mubarak Shah[1,2], Benjamin Yao[2], and Trishul Chilimbi[2]

[1] Center for Research in Computer Vision, University of Central Florida
rohit.gupta@ucf.edu, shah@crcv.ucf.edu
[2] Amazon
{jayunn,atawari,sontran}@amazon.com

Abstract. Pre-trained vision-language models (VLMs) have enabled significant progress in open vocabulary computer vision tasks such as image classification, object detection and image segmentation. Some recent works have focused on extending VLMs to open vocabulary *single label* action classification in videos. However, previous methods fall short in holistic video understanding which requires the ability to *simultaneously recognize multiple actions and entities* e.g., *objects* in the video in an open vocabulary setting. We formulate this problem as open vocabulary *multi-label* video classification and propose a method to adapt a pre-trained VLM such as CLIP to solve this task. We leverage large language models (LLMs) to provide semantic guidance to the VLM about class labels to improve its open vocabulary performance with two key contributions. First, we propose an end-to-end trainable architecture that *learns* to prompt an LLM to generate *soft attributes* for the CLIP text-encoder to enable it to recognize novel classes. Second, we integrate a temporal modeling module into CLIP's vision encoder to effectively model the spatio-temporal dynamics of video concepts as well as propose a novel regularized finetuning technique to ensure strong open vocabulary classification performance in the video domain. Our extensive experimentation showcases the efficacy of our approach on multiple benchmark datasets.

Keywords: Open Vocabulary · Multi-Modal · Video Understanding

1 Introduction

Video classification is a critical challenge in computer vision. Its goal involves recognizing a diverse array of concepts depicted in a video, which may include primarily static entities such as objects and scenes, as well as dynamic actions. In the classic setting, the vocabulary of all possible classes of interest is known in advance, and the model is trained in a supervised manner using a labeled dataset. The labor-intensive process of manual annotation often results in video datasets that are narrowly focused, such as those limited to specific sports or simple activities, which restricts the breadth of concepts models can learn. As video

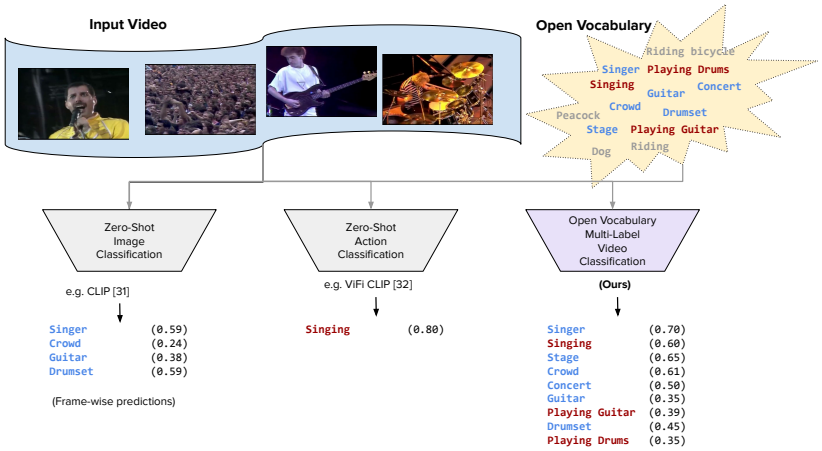


Fig. 1: Our task aims to recognize multiple classes in a video from an open vocabulary provided at inference time, including **entities** (in blue) such as objects and scenes, and **actions** (in red). Prior zero-shot *image* classification approaches such as CLIP can label the salient entity in each frame (left), while zero shot action-recognition approaches (e.g. ViFi-CLIP) (middle) can classify the main *action* in the full video. Our method (right) can recognize all action or entity classes present in the video.

applications are becoming widespread, there is an increasing need for developing video models that can recognize a broad range of concepts. This necessitates the development of open-vocabulary approaches for video classification that can recognize a diverse range of video concepts, including those that were not part of the class vocabulary present in the labeled training dataset. In this work, to solve this problem, we aim to leverage recent advances in vision-language models (VLM), which are trained to align visual and textual data and large language models (LLM), which have been demonstrated to possess a rich understanding of the world due to large scale pre-training.

VLMs that are trained for image-language alignment on image-text pairs at large-scale exhibit remarkable zero-shot visual recognition performance across a diverse set of tasks [13,22,34,37]. For classification tasks, the pretrained text encoder acts as a "*label encoder*" for the class labels, mapping them into the same representation space as the the visual embedding of the image, which allows for the image to be classified by ranking the labels in order of representation similarity. Since VLMs are primarily trained to rank, picking the top match usually results in excellent performance on single label zero-shot classification. However, merely ranking the labels is insufficient to achieve open-vocabulary multi-label classification, and as a result, specialized methods [46] have been developed for multi-label image classification using VLMs. VLM based zero-shot image classification has been further improved by LLM-based prompting, in which LLMs are used to generate descriptive class definitions which are then used as prompts to the VLM text encoder during inference [32,36]. Other works have adapted VLMs for zero-shot *single-label* action classification in videos [49,50]. A typical video, however, contains multiple concepts including objects, actions, scenes, events, etc. Our goal is to develop a video classifier that can simultaneously recognize

the presence of these different concepts from any vocabulary specified at inference. We refer to this task as open-vocabulary multi-label video classification. An illustration of the subtle differences between the prior tasks and our proposed task is presented in Fig. 1.

Open-vocabulary multi-label video classification presents two unique challenges. First, unlike the single label open-vocabulary setting, in the multi-label setting, identifying relevant concepts in a video based on VLM similarity score performs poorly, as VLM similarity scores for different types of concepts (e.g., actions, objects) often fall within different ranges, even with LLM-guided prompting (see Section 4.6).Addressing this issue requires end-to-end finetuning of the VLM with LLM guidance on datasets with diverse video concepts. However, performing end-to-end finetuning while simultaneously incorporating the benefits of LLM-guided prompting, presents additional technical challenges, particularly in backpropagating gradients through the VLMs text encoder's tokenizer. Second, adapting pretrained image-language models to recognize video concepts while retaining strong zero-shot performance is challenging, as the image-language pretraining datasets are orders of magnitude larger than the largest labeled video datasets. VLM adaptation tends to overfit to the video data easily, thus losing their generalization capability [49].

In this work, we attempt to address these challenges for open vocabulary multi-label video classification. First, to simultaneously recognize multiple video concepts in open-domain, we finetune the VLM in an end-to-end manner with LLM guidance to ensure proper ranking between different types of video concepts. Particularly, we extend the LLM-guided prompting approach for open-vocabulary classification by introducing learnable prefixes for prompting the LLM. To directly utilize the LLM output representations in the VLM's text encoder for end-to-end training, we propose a prompt transformer. This prompt transformer, in conjunction with the learnable prefixes, not only facilitates the integration of the LLM's world knowledge into the VLM's text representation but also aids in mitigating the discrepancy between VLM scores for various types of concepts. Second, for the effective recognition of temporally varying video concepts, we integrate an auxiliary temporal modeling branch with the VLM's vision encoder [29] and introduce a novel regularization penalty to retain the VLM's zero-shot performance during the adaptation to videos. Through extensive experimentation and analysis we demonstrate that our proposed approach significantly outperforms baseline solutions obtained by extending prior work on single label open vocabulary image and video classification.

In summary, in this work we make three major technical contributions:

- A novel end-to-end trainable approach to learn a strong label encoder for open vocabulary multi-label video classification by adapting an LLM to learn to prompt a VLM's text encoder.
- An approach for enhancing pre-trained VLM image encoders with temporal modeling capability while also retaining strong open-vocabulary performance.
- Defining a new benchmark for open vocabulary multi-label video classification, consisting of 2 closed vocabulary and 3 open vocabulary evaluation datasets,

by leveraging existing datasets. Our approach is benchmarked against 6 different strong baselines, and significantly outperforms them.

2 Related Works

Vision-Language Representation Learning: Recently, image-language models [14, 21, 37, 45, 56, 58] have drawn huge attention because of their effectiveness in learning generic visual representations that are transferable to several downstream tasks like classification, retrieval, etc. This success can partly be attributed to the recently available large-scale image-text datasets [6, 41–43, 47]. However, this is not the case for the video data. Therefore, to perform video-language pretraining, most recent works [8, 9, 31, 35, 54] bootstrap from a pretrained image-language model and then perform some form of lightweight adaptation on the video datasets. Our work is orthogonal to these works in the sense that we attempt to improve the open-vocabulary video classification performance of these methods instead of trying to learn better generic visual representations.
Open-Vocabulary Classification: Even though vision-language models excel at generic visual understanding tasks, they do not perform optimally in visual recognition tasks out of the box. To improve the zero-shot open-vocabulary visual recognition capability of the vision-language models, the recent works either resort to regularized finetuning [50, 52] or prompt learning [46, 49, 61]. However, these works either focus on solving single label tasks or only perform adaptation for image recognition tasks. In sharp contrast to these works, in our work we focus on open-vocabulary video classification task.
Large Language Models in Vision: Recently, Large Language Models (LLM) have been utilized in vision tasks primarily because of their excellent in-context and zero-shot learning performance especially in commonsense reasoning tasks [17]. To be particular, LLMs have been used to rewrite noisy ASR text [44], to generate classification labels by mapping textual description to a predefined task list [25], extracting verbs from textual descriptions to improve action understanding [24]. LLMs have also been used to initialize textual encoders for vision language encoder [60]. In some recent works, vision representations have been aligned with LLM input space to solve multitude of vision-language tasks [20, 27]. These multi-modal models have also been used to generate captions for images which are then utilized to train better vision-language models [7].
Prompting Vision-Language Models with LLMs: Some prior works have leveraged LLMs to augment the open vocabulary understanding capabilites of Vision-Language models. Classification By Description [32], CuPL [36], LLM guided concept bottleneck [55] all utilize LLM generated class descriptors to prompt CLIP to improve image classification, and make it more interpretable. WaffleCLIP [40] demonstrated that using random prompts for CLIP and LLM generated higher level concepts to describe datasets can also improve classification. LLM + VLM based approaches have also been extended to object detection [15] and point cloud understanding [62]
Parameter efficient fine-tuning of LLMs: As full parameter finetuning of LLMs is not practical, quite a few parameter efficient finetuning methods

have been developed. One of the earliest approaches utilized Learnable Soft prompts [19], this is the approach we utilize to guide the LLM in our method. Similar approaches such as Low Rank Adaptation (LoRA) [12], Prefix-Tuning [23] and P-Tuning [30] could also be adopted in principle.

3 Method

Our proposal to boost the open vocabulary multi-label video classification capabilities of CLIP consists of two key parts: first, an end-to-end trainable label encoder leveraging an LLM for strong open vocabulary capabilities, and the CLIP text encoder for visual alignment. The label encoder consists of a frozen LLM guided with learnable prefixes, whose outputs are mapped to the frozen CLIP text encoder using a learnable prompting transformer. Second, we enhance the CLIP image encoder's capabilities for understanding temporal dynamics. This is achieved by using a lightweight temporal modeling branch to enhance the CLIP image encoder. The details of our approach are illustrated in Figure 3. Each part of our method is discussed in detail in this section.

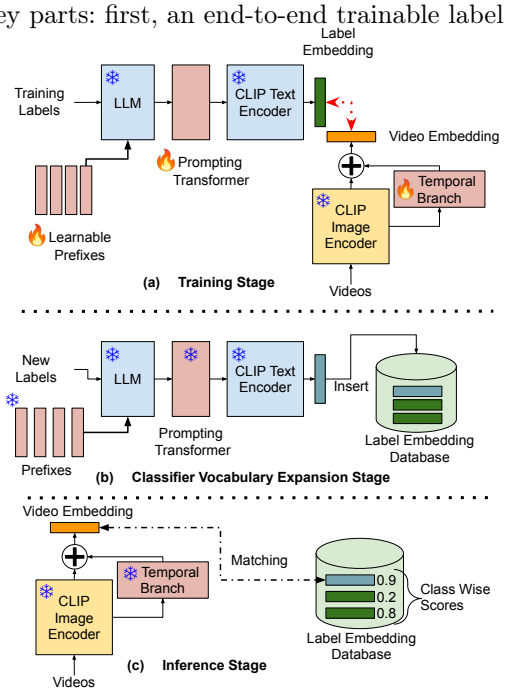


Fig. 2: Our open vocabulary classification method includes three stages of operation. During the **(a) training stage**, we train our label and video encoders on closed set training labels. New class labels can be added to the vocabulary after training by employing the **(b) classifier vocabulary expansion stage**. During the **(c) inference stage** video embeddings are computed and matched with the label embeddings database to get the classification scores.

An overview of our approach is provided in Fig. 2. Our method has three broad stages. Firstly, during the training phase, both the label encoder and video encoder are trained simultaneously, as shown in (a).

Secondly, during the classifier vocabulary expansion stage, embeddings for class labels are calculated and saved into a label embedding database, as shown in (b). This vocabulary can be extended at any point after training, thus allowing our method to be used in the open vocabulary setting. Finally, during the inference stage, video features are computed and compared against label embeddings from the database. As the label embeddings are pre-computed, the computational overhead during inference over standard CLIP models is minimal.

3.1 LLMs for Semantic Enrichment of Label Embeddings

Pretrained VLMs have strong open-vocabulary image classification performance owing to their large scale image-text pretraining. LLMs on the other hand are trained on even larger scale of text data with training objectives that enable a rich semantic understanding. As a result, compared to LLMs, VLMs have a more limited semantic understanding of natural language and have difficulty understanding concepts such as the relationship between different class labels. For instance, prior works have found that while VLMs are able to classify ImageNet images at a fine-grained level, their accuracy at higher levels of the class hierarchy is poor [53]. E.g., while they may recognize an image of a `lion`, they are unable to understand that a `lion` is also a `feline`, a `mammal` and an `animal`. However, an LLM is very effective in comprehending the hierarchical relationship between these labels, among many other capabilities, due to the large-scale training. Hence, in this work, we utilize an LLM to generate complementary information about the class labels that can be utilized to improve the vision-language alignment of VLM leading to better open-vocabulary classification performance.

Fixed LLM prompting. As a baseline, we first develop an LLM-based prompting method to adapt CLIP for open vocabulary video classification, extending the approach of Menon & Vondrick [32] for open vocabulary image classification. We design a prompt template (see implementation details in Section 4.3) for an encoder-decoder based LLM with the class name and a question asking it to generate useful features for visually distinguishing that class. The LLM output is then parsed into a list of textual descriptions, hereafter referred to as attributes. These attributes along with the class name is used to prompt the CLIP text encoder and the resultant text-embeddings are mean-pooled to obtain an attribute enriched text-embedding for the class. We mean-pool CLIP vision embedding of video frames to generate video embeddings and then perform open vocabulary classification by matching video embeddings with the text-embeddings of different classes. However, this simple baseline has a crucial weakness: it is not trainable, as there is a need to de-tokenize and tokenize the LLM text output to CLIP, which doesn't allow the flow of gradients. As a result, the process of generating class attributes by prompting an LLM cannot be improved by training on a labeled video dataset. To remedy this, we propose an end-to-end trainable architecture that integrates the LLM with the CLIP text encoder.

End-to-end learnable LLM prompting. Our proposed architecture, summarized in Figure 3, incorporates a learnable prompting framework with the frozen LLM to generate the inputs to a frozen CLIP text-encoder. The learnable components on the text side are limited to N learnable prefixes/vectors to the prompt template used for querying the LLM and a prompt transformer that transforms the sequence of tokens from the LLM to input soft prompts for the CLIP text-encoder. In our implementation we avoid the discrete and non-differentiable operations like detokenization at the LLM decoder output and re-tokenization at the CLIP text-encoder input. This allows the prompt transformer to directly connect the LLM output semantic space to the CLIP input

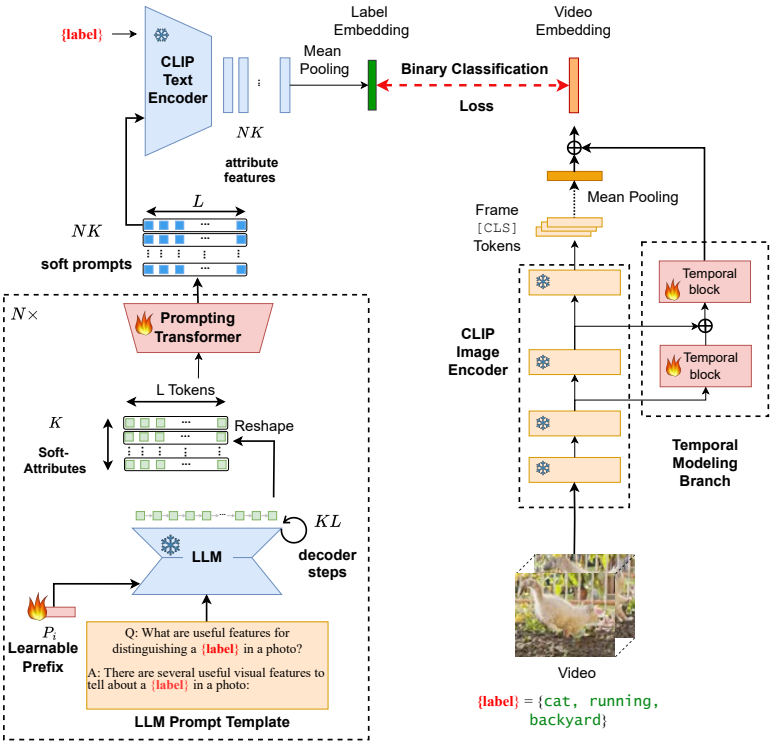


Fig. 3: Our end-to-end trainable system for open vocabulary video classification. The class labels are used by the LLM to generate useful class attributes for the CLIP text encoder which provides a visually aligned label embedding. The learnable input prompts to the LLM guide it to generate soft-attributes useful for video classification. Our prompt transformer learns to map from the LLM output space to the CLIP input space. In order to bootstrap video understanding to CLIP's vision encoder we add additional spatio-temporal modeling layers. Details about each of these components are provided in Section 3.

semantic space, ensuring that the entire text model is differentiable and therefore end-to-end learnable. We describe the details below.

For each of the N LLM prefixes, we first construct an input sequence to the LLM comprising of the prefix followed by a fixed prompt template that contains the label name ℓ from the input data sample (see Figure 3). The text input to the LLM is mapped using a tokenizer and the LLM's embedding layer into a sequence of tokens $I \in \mathbb{R}^{M \times d}$, where M and d represent number of input tokens and the dimension of the embedding space respectively. All our N learnable prefixes are d-dimensional vectors. We use $P_i \in \mathbb{R}^d$ to represent the i-th prefix. Each prefix is concatenated with the tokens of the prompt template, yielding a unified sequence of tokens $[P_i; I] \in \mathbb{R}^{(1+M) \times d}$. This combined sequence is then processed through the frozen encoder-decoder layers of the LLM. For each prefix, we run KL decoding iterations of the LLM decoder to generate a sequence of KL decoded tokens. These tokens represents useful class features in the LLM's

semantic space. As the prompt templace specifically asks LLM to output features as a list, we split the sequence of tokens evenly into K subsequences of L tokens each. Due to the nature of LLM prompt template, which prompts the LLM to output useful visual features, we refer to the subsequences of tokens as *soft attributes*, as they are sets of continuous vectors instead of discrete attributes in natural language. The K soft attributes are then individually processed by the prompt transformer to generate K *soft prompts* to the CLIP text encoder. Repeating this operation for each of the N LLM prefixes, we get NK soft prompts. Each soft-prompt is then concatenated with the tokenized label embedding and then processed by the frozen CLIP text-encoder to generate an attribute feature. All NK resulting features are mean-pooled and normalized to obtain the final label embedding $f_t(\ell)$, where ℓ is the label name.

3.2 Regularized Parallel Temporal Modeling

The simple image transformer of the CLIP vision encoder does not model the temporal dynamics of the video, which is essential to enhance the ability to recognize entities in videos. We enhance our vision encoder by adding a parallel temporal modeling branch to the last T layers of the CLIP vision encoder as illustrated in Figure 3. We freeze the CLIP vision branch and train only the newly added temporal layers. Each block in the newly added branch consists of a spatial attention layer initialized from their corresponding CLIP weights, and a temporal attention layer that is randomly initialized. All frames of the video are processed independently by the CLIP vision backbone. Then, at the first temporal modeling block, the temporal token TMP is created by averaging the CLS tokens across frames, meanwhile learnable spatial and temporal positional embeddings are added to each patch token. The tth temporal modeling block takes in as input a weighted combination of the previous temporal modeling block and the corresponding layer in the CLIP backbone. Symbolically if $V_t^{\intercal}$ represents the patch tokens from the temporal block at the tth block and V_t represents the patch tokens for the corresponding CLIP layers, the patch tokens for the tth block $V_t^{\intercal} = V_{t-1}^{\intercal} + Proj_{spatial}(V_s)$, where V_s is the corresponding token from the CLIP backbone. After that the tokens are processed by the divided space-time attention layer and spatial layer. The overall video embedding is generated at the end by mean pooling the final TMP token and the CLS tokens for each frame from the CLIP backbone.

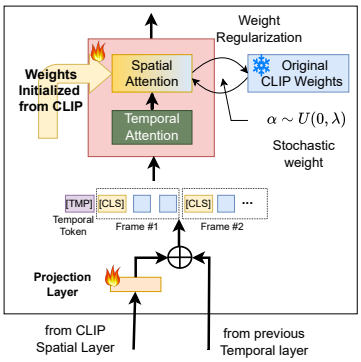


Fig. 4: Our Temporal Block takes in frame patch tokens from the CLIP image encoder, projecting and then fusing them with the temporal branch tokens from the previous block. The temporal token (TMP) is enriched with the CLS tokens from all frames. Divided Space-Time attention layers form the core of the block. Weights for the spatial attention layer are initialized from CLIP weights and regularized using our stochastic weight averaging scheme.

A key drawback of adding a deep parallel branch is that while it improves performance in the finetuned setting, it diminishes CLIP's zero shot performance. In order to overcome this limitation in the zero-shot setting, we propose to use weight regularization on the spatial attention layers. The weight regularization operation tries to match predictions made by the current set of finetuned weights and a randomly weighted average of the current weights and the original CLIP weights. Symbolically, at each iteration we use weights θ which are a stochastic weighted average of the finetuned and the original frozen weights as given by:

$$\theta = \alpha\theta_{ft} + (1-\alpha)\theta_{frozen}, \text{ where } \alpha \sim U(0, \lambda)$$

We observe that an empirical value of $\lambda = 0.5$ works well across datasets.

Relationship to prior works: Prior approaches to temporal modeling have taken two different directions: the first set of approaches has focused on improving performance in the finetuned setting, whereas the second set of approaches has primarily focused on preserving CLIP's zero shot performance. STAN [29] is a state-of-the-art finetuning focused approach which adds a parallel temporal modeling branch which is able to leverage both high-level and low-level features from the CLIP vision encoder. The divided space-time attention in STAN's temporal block is inspired by the TimeSformer [2] architecture. We differ from STAN in order to achieve stronger open vocabulary performance in a number of ways. Firstly, unlike STAN we do not finetune the CLIP image encoder, only the added parallel branch. Additionally, we propose stochastic weight regularization to prevent the temporal branch from drifting too far from the original CLIP feature space. Our ablation experiments (Table 3) demonstrate that both these choices significantly improve our performance. Some form of weight regularization has also been explored by methods such as Open-VCLIP [51]. However, Open V-CLIP is designed only to take in 3 frames at a time, and its temporal modeling range is limited. Our approach is able to achieve the best of both approaches.

3.3 Training objective

We train our model on multi-label video datasets. To construct a training batch, we first sample a set $\mathcal{B}$ of B videos from the dataset. Each video v has associated with it a set of positive class labels which we denote $\mathcal{P}(v)$. In addition to the positive class labels for the sampled videos, we augment the batch with random negative class labels for each video. To obtain the negatives, we first identify the set $\mathcal{P}_\mathcal{B} := \cup_{v\in\mathcal{B}}\mathcal{P}(v)$ of distinct classes among all the positive labels from all the videos in the batch. We then sample a random set $\mathcal{N}_\mathcal{B}$ of $4B - |\mathcal{P}_\mathcal{B}|$ classes from the rest of the class vocabulary of the dataset. For each video v, we then choose all non-positive classes from $\mathcal{P}_\mathcal{B} \cup \mathcal{N}_\mathcal{B}$ as negative labels. We use $\mathcal{N}(v) := (\mathcal{P}_\mathcal{B} \cup \mathcal{N}_\mathcal{B}) \setminus \mathcal{P}(v)$ to denote the set of all negative labels for v. The resulting positive and negative video-label pairs are then treated as training samples for binary classification. Each training sample is a label, video pair (ℓ, v) where ℓ represents the label name and v the video. With this batch construction, the total number of training samples in a batch could be variable, but the total number of videos is fixed at B and the total number of classes present among the samples is fixed at $4B$.

For each training data sample (ℓ, v), the text and video encoders respectively generate a unit-norm text embedding $f_t(\ell) \in \mathbb{R}^D$ and a unit-norm video embedding $f_v(v) \in \mathbb{R}^D$ where D is the common embedding dimension. The score $s(\ell, v)$ for this data sample is given by the inner product:

$$s(\ell, v) = (f_t(\ell))^\intercal f_v(v). \tag{1}$$

The model is trained with a weighted binary cross entropy loss:

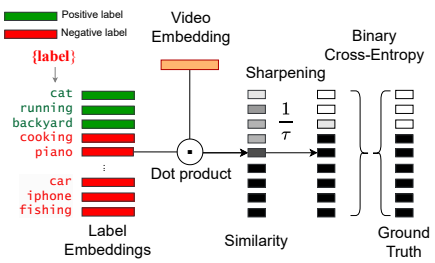


Fig. 5: Our Binary Classification Loss consists of binary cross-entropy applied on top of predicted Video-Label feature similarities sharpened through the use of a temperature scaled sigmoid operation.

$$\mathcal{L}(\mathcal{B}) = -\sum_{v \in \mathcal{B}} \left[\sum_{\ell \in \mathcal{P}(v)} \log p(\ell, v) + w \sum_{\ell \in \mathcal{N}(v)} \log(1 - p(\ell, v)) \right] \tag{2}$$

where $p(\ell, v) := \sigma\left(\frac{s(\ell,v)}{\tau}\right)$, τ is a temperature parameter, $\sigma(.)$ is the sigmoid function and $w > 0$ is a weight hyperparameter.

4 Experiments and Results

4.1 Datasets

In order to train our model on a wide range of concepts, we train it on a mix of YouTube8M [1] and Kinetics-400 [16]. YouTube-8M (YT-8M) is primarily labeled with entities, whereas Kinetics-400 (K400) is labeled only with actions.

As YT-8M is composed of a random sample of the *whole* of YouTube, a significant portion of video its samples are from video game streams. Many of these videos are just tagged with the video game title as the label irrespective of the actual entities and actions present in the video. A significant portion of its label vocabulary is dedicated to video game titles (771 out of 3862 total) are dedicated to video game titles. We observe training instabilities due to this, and to fix this issue, we remove video game titles from the *training* vocabulary. We also remove some of the least frequently occurring labels during training to reach a training vocabulary of 2429 classes. This lead to stabilized training without further issues. Note that for closed set evaluation on YT-8M, we use the YouTube-8M Segments validation set, which contains human verified labels for 1000 classes. K400 on the other hand is widely used, and has a clean vocabulary and high quality human verified action labels. From the training perspective, the only limitation is the absence of entity (object, scenes etc.) labels.

For evaluation, we use 3 open-vocabulary test datasets: TAO (Tracking Any Object) [5], ActivityNet [11] and RareAct [33]. TAO and ActivityNet are exclusively object and action focused datasets respectively. RareAct has object, action labels, and focuses on unusual object-action combination. TAO dataset was developed for evaluating object trackers, however analogous to how object

detection datasets like MS-COCO are used for evaluating multi-label image classification methods such as DualCoOp [46], we transform TAO into a multi-label video classification dataset by ignoring localization annotations. ActivityNet often has multiple actions in a given video. For our evaluation each annotation from RareAct provides three labels: object, action and the (unusual) object-action combination, this provides an interesting test of the open vocabulary capabilties. Altogether, these five datasets provide a comprehensive evaluation of a model's closed and open vocabulary video classification performance.

4.2 Baselines

As there are no prior works that explicitly address multilabel open vocabulary video classification, we extend the following methods from the zero shot image and action classification literature to our multi-label video classification setting for comparison.

CoOp [61]: Short for **Co**ntext **Op**timization, CoOp learns prompts for the CLIP text encoder as a lightweight adaptation technique for classification.

DualCoOp [46]: Is an extension of CoOP to the multi-label setting where both positive and negative prompts are learnt for the CLIP text encoder. For a given label, prediction is based on whether the positive or negative prompted version has higher similarity with the image feature.

LLM + CLIP (Frozen): This baseline was discussed in Section 3.1 and is illustrated in Figure 8 of Supplementary.

ViFi-CLIP [38]: In this baseline, there is no temporal modeling, and both the CLIP image and text encoder are finetuned on the training dataset.

As `CoOp` and `ViFi-CLIP` were developed for single label classification, to utilize them in our setting we replace their contrastive loss function with our multi-label classification loss. We also drop the region aggregation aspect of `DualCoOp` and only test the dual prompting architecture. We refer to these baselines as **CoOp***, **DualCoOp*** and **ViFi-CLIP*** to reflect these differences.

Method	Closed-Vocabulary		Open-Vocabulary		
	YouTube-8M	Kinetics	TAO (Entities)	ActivityNet (Actions)	RareAct (Entities+Actions)
Frozen CLIP-based Methods					
CLIP with Class name Prompt	6.3	26.2	43.8	44.2	9.5
CLIP with Prompt Templates	6.8	30.5	46.0	45.9	11.4
CLIP with Fixed LLM Prompts	6.9	30.6	50.2	46.8	11.5
Trainable Baseline Methods					
CoOp	2.7	17.8	35.0	28.8	3.5
DualCoOp	8.3	23.9	47.1	33.0	7.6
ViFi-CLIP	3.4	10.9	58.3	17.2	4.1
Ours					
CLIP + Learnable LLM Prompts	9.4	32.8	51.4	47.1	11.9
+ Temporal Modeling	14.8	42.0	63.8	47.1	12.4
+ Synthetic Labels	**16.7**	**43.2**	**65.5**	**50.2**	**13.2**

Table 1: AUPR scores for all methods on all datasets.

4.3 Implementation details

We use Open AI CLIP-B/32 [37] as the backbone VLM and Flan-T5-XL [4] as the promptable LLM. We sample 8 frames per clip during training, and during evaluation we use 4 clips per video (total of 32 frames). Our best model uses 4 learnable LLM Prompts, and 4 layers of temporal modeling. (Following the notation from the method section, $N = 4$, $K = 5$, $L = 5$ and $T = 4$). Following the Flan-T5 text generation instructions we use the following prompt:

```
Q: What are useful features for distinguishing a {label} in a photo?
A: There are several useful visual features to tell about a {label}
in a photo:
1. <extra_id_0>
```

Where {**label**} represents a given class label and `<extra_id_0>` is T5 decoder's start token ID. In case of the frozen baseline, when presented with this prompt, the LLM produces text that consists of a list of label attributes, which can be parsed and separated into different prompts for CLIP. We mimic this approach in our learnable version by chunking LLM output into K groups of L tokens each.

We use a batch size of 12 videos/GPU across 32 A100 GPUs (Total Batch Size=384) and train all models for 30000 training steps and evaluate at 10000, 20000 and 30000 steps, providing the best results for each methods across these 3 checkpoints. We do this to ensure a fair comparison as our different baselines and methods have widely varying number of trainable parameters, it would not be a fair comparison to use the same number of steps for each.

4.4 Metrics

We report two sets of metrics to evaluate the performance of the models in the open vocabulary setting. Area Under Precision-Recall curve (AUPR) summarizes the overall classification performance across the entire precision-recall trade-off. Secondly, we report the Peak F1-Score for the model on each dataset, which is the F1-Score achieved by that model at the optimal threshold chosen by an oracle for that dataset. This metric captures the classification performance obtained by an open vocabulary classification method on a dataset, if the threshold alone could be tuned, e.g., by using a labeled validation set.

4.5 Results

AUPR scores for each method are presented in Table 1 and Peak F1 scores are presented in Table 2. Our learnable LLM prompting approach outperforms both frozen and trainable baselines across both closed-vocabulary and open-vocabulary classification tasks. Also notable is the enhancement in performance due to our temporal modeling approach (average gain of $\approx 5\%$ in AUPR). As previously mentioned in our discussion of the dataset, the training datasets YT-8M and K400 have certain limitations (in particular, YT-8M has few action labels and K400 has no object/entity labels). We present further improved results obtained by augmenting our training dataset with additional class labels obtained using a multimodal LLM (pipeline is detailed in the Supplemental). As seen from the last rows of Tables 1 and 2, this approach gives a further $\approx 2\%$ improvement, and is specially effective for actions.

Method	Closed-Vocabulary		Open-Vocabulary		
	YouTube-8M	Kinetics	TAO (Entities)	ActivityNet (Actions)	RareAct (Entities + Actions)
Frozen CLIP-based Methods					
CLIP with Class name Prompt	14.9	34.2	44.6	47.1	17.6
CLIP with Prompt Templates	20.8	36.8	49.2	50.1	20.5
CLIP with Fixed LLM Prompts	21.6	37.3	50.2	51.4	19.8
Trainable Baseline Methods					
CoOp	5.8	25.5	43.9	35.5	10.5
DualCoOp	16.2	33.2	49.0	40.5	15.0
ViFi-CLIP	5.2	19.3	54.4	24.7	9.6
Ours					
CLIP + Learnable LLM Prompts	23.6	42.4	52.8	51.1	22.6
+ Temporal Modeling	31.5	46.2	**59.6**	52.6	24.3
+ Synthetic Labels	**32.7**	**46.6**	56.6	**53.8**	**25.1**

Table 2: Peak F1 scores for all methods on all datasets.

(a) Closed Vocabulary Evaluation Datasets. **(b)** Open Vocabulary Evaluation Datasets.

Fig. 6: F1-Scores at different thresholds for closed and open vocabulary evaluation datasets. Our end-to-end trained model can achieve better performance across datasets with a single threshold (labeled by gray vertical line) chosen on the supervised datasets.

4.6 Improved score calibration across datasets

In order to use a classifier in a truly open vocabulary setting in practice, the classification score needs to be well calibrated across different types of concepts. This would allow us to pre-select an optimal threshold to generate binary classification for all concepts. In Figure 6 we demonstrate the robustness offered by our method in setting a threshold that works for a wide range of concepts, and the advantage offered by our method over the frozen CLIP + LLM baseline. These figures plot the F1 scores for multi-label classification across thresholds provided by both methods with Figure 6a showing the numbers on the validation splits of the training datasets, and Figure 6b showing the numbers in the open vocabulary setting on two diverse datasets. A reasonable choice for a classification threshold would be to pick the threshold that maximizes the minimum validation F1-score for the datasets used in training. The resulting choice of thresholds for our method and for frozen CLIP are shown in Figure 6a. We see that for the baseline method the performance on TAO is poor, while that on ActivityNet is modest. However with our proposed method the F1 scores for both evaluation datasets are simultaneously high. This result empirically validates the versatility that is achieved by our learnable LLM-based prompting scheme over the baseline, which makes it possible to pre-select a threshold for using the solution as a black box classifier for all unseen concepts.

	Open Vocabulary	
# Blocks	Objects	Actions
5	65.4	48.1
4	**65.5**	**50.2**
2	64.6	49.1
1	62.3	47.7

(a) Effect of Temporal Modeling

		Open Vocabulary		
Spatial Reg.	Backbone	Objects	Actions	Geo. Mean
✓	❄	56.6	53.8	**55.2**
✗	❄	62.1	32.5	44.9
✗	🔥	42.7	28.3	34.8

(b) Effect of our Regularization

Table 3: Ablations of Temporal Modeling Branch Architecture and Regularization

Label Encoder			Open Vocabulary	
LLM	PT	CLIP	Objects	Actions
Learnable Prompt	✓	✗	25.2	8.5
Fixed Prompt	✗	✓	62.5	41.3
Fixed Prompt	✓	✓	64.9	48.1
Learnable Prompt	✓	✓	**65.5**	**50.2**
✗	✗	✓	60.1	34.5

Table 4: Ablating the Label Encoder. PT → Prompt Transformer

4.7 Ablations

We scientifically ablate each part of our framework to demonstrate improvements provided by each. For the sake of brevity, we only carry out ablation evaluations on TAO (Objects) and ActivityNet (Actions) and report AUPR.

- **Temporal Modeling Blocks:** We find that using more Blocks improves performance, until around 4. (Table 3a)
- **Temporal Modeling Weight Regularization:** We find that using our weight regularization strategy for the spatial layers of the temporal branch prevents the model from overfitting to certain concepts. We also validate our choice of keeping the CLIP vision backbone frozen and find that unfreezing the backbone leads to severe over-fitting. (Table 3b)
- **Label Encoder:** To demonstrate the efficacy of using an LLM with learnable prompts and the CLIP text encoder together as the label encoder, first, we show that using LLM without CLIP text encoder leads to a significant drop in performance. This demonstrates that the visual-text alignment learned by CLIP is essential. Next we demonstrate the benefits of the learnable prompts over using a fixed prompt. For completeness we also provide results with the LLM completely removed (Table 4).

5 Conclusion

We introduced the problem of open vocabulary multi-label video classification and proposed a solution leveraging LLMs and pre-trained VLMs. Through our extensive experiments we demonstrated strong performance on both actions and entities with a single model. Our proposed approach benefits from two key innovations. First, we proposed a method to prompt VLMs more effectively by utilizing the LLM output representations through a trainable prompt transformer and learnable LLM prompts. Second, we introduced a temporal modeling architecture and a regularized finetuning approach to improve the video understanding capability of the vision encoder while retaining strong open-vocabulary performance. Our ablations validate the efficacy of each of these contributions.

Supplementary Material for *Open Vocabulary Multi-Label Video Classification*

Overview

This supplementary material is organized into the following sections:

- Section A Comparison of our results with Supervised State of the Art
- Section B Comparison of our results with Single Label Open Vocabulary Baselines
- Section C Comparison of our results with a Multi-Modal LLM
- Section D: Evaluation of our approach on Single Label Classification tasks
- Section E: Evaluation of our approach on EgoCentric tasks
- Section F: Additional Ablations for Label Encoder
- Section G: Additional Ablations for Temporal Encoder
- Section H: Inference and training costs of our approach
- Section I: Further details about the Synthetic Label Pipeline
- Section J: Further details about all the Baselines reported
- Section K: Additional implementation details
- Section L: Qualitative Results

A Comparison with Supervised SOTA

In order to provide some additional context for our results, we also evaluate some existing state of the art baselines on our downstream datasets.

The best ActivityNet trained model with public weights is ASM-Loc (He et al. [10], CVPR 2022). Our open vocabulary classifier comes within 10% Peak F1-Score of this supervised model despite not being trained on any ActivityNet data. We also provide results finetuning ViFi CLIP on downstream datasets, which diminishes open vocabulary generalization capabilities. Our single model is competitive across both datasets.

Table 5: Comparing with Supervised Results (Peak F1-Score)

Method	TAO	ActivityNet	Geometric Mean
Ours (Zero-Shot)	56.6	53.8	**55.2**
Supervised Methods			
ASM-Loc CVPR 2022	-	63.1	-
ViFi-CLIP (TAO-FineTuned)	60.2	12.6	27.5
ViFi-CLIP (ActivityNet-FT)	32.7	58.2	30.4

B Comparison with Single Label Open Vocabulary Baselines

STAN [29] is intended for fully fine-tuned setting; it doesn't report any zero-shot results. Open V-CLIP [50] is trained to solve single label classification. In contrast, our goal is to perform multi-label classification in the zero-shot setting. For comparison we provide zero-shot results for both. As STAN doesn't provide pretrained weights, we train it on our training dataset. For Open V-CLIP, we use author provided pretrained weights.

Table 6: Open Vocabulary baselines (Peak F1)

Method	TAO	ActivityNet
`STAN` (K400+YT8M, ours)	58.1	27.6
`Open V-CLIP` (original)	43.9	50.2
Ours	**59.6**	**52.6**

C Comparison with Multi-Modal LLMs

Recently in the literature [57,59], general purpose multi-modal LLMs have been demonstrated to achieve competitive performance across a range of video understanding tasks. They are not practical for our setting, since they impose a significant computation cost, however to demonstrate the advantage of our solution over multi-modal LLMs, we construct two LLaVA-based inference baselines. For the first, we prompt LLaVA regarding the presence of a class label in video frames. For second, we closely follow our synthetic label generation pipeline (see Section I) and generate frame captions using LLaVA. The captions are then classified using CLIP's text encoder. The results in Table 7 show that LLaVA performs significantly worse than our method, even when no synthetic labels are used for training.

Both LLaVA based approaches require running the Multi-Modal LLM for every video at inference. Additionally, for the first approach, we need to run it for every label in the validation vocabulary. In contrast, for our method the LLM is not used during inference, but only when a new label is added to the classification vocabulary.

D Zero-Shot Single Label Action Classification

Our open vocabulary model though trained for multi-label classification is also competitive (see row **(a)** in Table 8) on the zero-shot single label action classification task.

Table 7: LLaVA Inference baselines (Peak F1-Score)

Method	TAO	ActivityNet	Inference Time (1× A100)
LLaVA Classification (yes/no polling every class)	27.8	11.5	10 min+
LLaVA Captioning + CLIP Classification	47.2	34.7	10s
Ours	**59.6**	52.6	0.25s
Ours w/ added synthetic labels during training	56.6	**53.8**	0.25s

A key difference between our approach and prior single label classification works tailored to this problem is our use of binary classification losses, which is essential for multi-label classification but is not optimal for single label classification, which only requires ranking the labels. In order to match the setting of prior works, we also train our model on only Kinetics-400 using Cross-Entropy loss and provide results in row **(b)** to show that it can exceed prior work such as ViFi-CLIP [38].

Model	**UCF101**	**HMDB51**	**Kinetics600**
No video data used in training			
CLIP	61.7	37.5	63.5
CLIP + LLM	73.8	46.1	64.8
Trained on *YouTube8M + Kinetics400*			
(a) Ours	74.1	53.2	67.7
Trained on *Kinetics400*			
Vi-Fi CLIP *	77.5	51.8	71.2
(b) Ours (Cross-Entropy Loss)	**79.0**	**54.5**	**72.8**

Table 8: Results on single label action classification datasets. Top-1 Accuracy is reported for all datasets. * Results reported in ViFi CLIP [38]

E Zero-Shot Evaluation on EgoCentric tasks

We provide results for scenario classification on Ego4d and verb & noun identification for Epic-Kitchens (unseen kitchens).

Table 9: Egocentric (Peak F1)

Method	Ego4D	EK-unseen (Verbs)	EK-unseen (Nouns)
CLIP	45.3	16.5	32.8
CLIP+LLM	48.7	20.3	39.1
Ours	**51.9**	**22.1**	**40.5**

Table 10: LLM Adaptation Ablations (Peak F1-Score, 10k training steps)

Steps	LLM Adapter	LLM-VLM Connector	TAO	ActivityNet
10k	LoRA (r=2)	Prompting Transformer	57.9	49.4
10k	LoRA (r=4)	Prompting Transformer	58.0	46.9
10k	LoRA (r=2)	Linear	46.2	38.9
10k	Prompts	Linear	48.4	35.2
10k	Prompts	MLP	52.9	44.7
10k	**Prompts**	**Prompting Transformer**	**58.3**	**50.8**
50k	Prompts	Prompting Transformer	59.6	52.6

F Additional Ablations for Label Encoder

We conducted additional ablations for the LLM adapter and LLM-VLM connector, with results shown in Table 10. Due to time constraints, we trained for only 10k steps, nearing convergence. We find that LoRA saturates after rank 2 and performs worse than prompt learning for Zero-Shot Generalization. Our prompting transformer outperforms MLP & Linear connectors.

G Additional Ablations for Temporal Encoder

We designed an alternative version of our architecture with serial blocks instead of parallel and train the model again. The results (Table 11) indicate that parallel blocks outperform serial blocks. As our main goal is open vocabulary classification, our temporal ablations (Table 4) are focused on regularization and related aspects. Exhaustive temporal architecture ablations are beyond the scope of a single paper, and different aspects of temporal modeling have been studied previously ([3], [26], [39]).

Table 11: Temporal (Peak F1)

Temporal Adapter	TAO	ActivityNet
Serial (n=4)	53.2	41.5
Parallel (n=4)	**59.6**	**52.6**

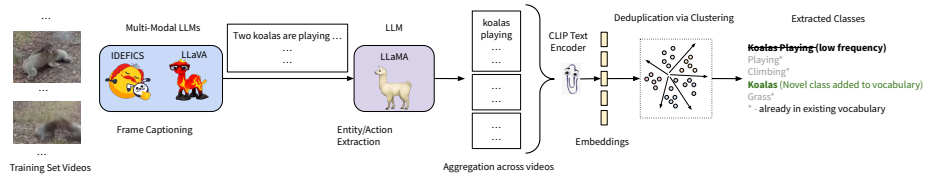
H Comparison of Computational Costs

Table 12: Computational Costs

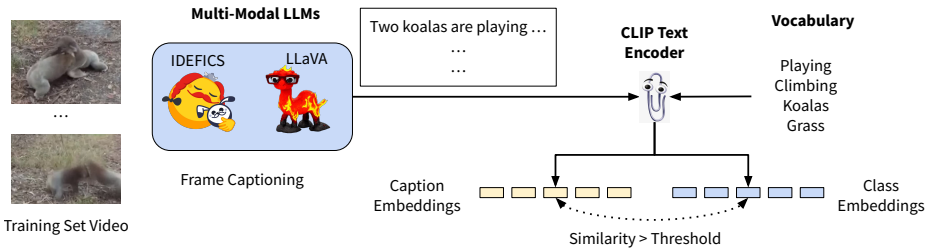
Method	**Training (YT8M+K400)**		**Inference time**
	Time	Mem/GPU	(batch size = 32)
ViFi-CLIP	36 Hrs	11.0 GB	338ms
Ours	40 Hrs	16.5 GB	393ms

Training time (on 16 × A100 GPUs) on YT-8M + K400 for our method is about 10% higher than ViFi-CLIP baseline. Inference on 1 RTX8000 is about 16% slower (batch size=32, using `torchinfo` package). Text embeddings for class labels can be pre-computed, only video features need computing on the fly during inference.

I Synthetic Label Generation Pipeline



(a) Expanding the label vocabulary.



(b) Assigning labels to videos.

Fig. 7: Incorporating synthetic labels into our training sets enhances our open vocabulary performance further. **(a) Vocabulary Expansion:** We have developed a pipeline to automatically extract action and object labels from a vast video dataset utilizing foundation models. For captioning video frames, we employ Multi-Modal Large Language Models (LLMs), specifically IDEFICS and LLaVA. Subsequently, LLaMA is prompted to distill object and action class labels from these captions. We aggregate these labels across videos and remove duplicates through clustering, forming a classification vocabulary. **(b) Label Assignment:** Labels from the vocabulary are aligned with the generated video captions using the text encoder from CLIP.

As illustrated in Figure 7 our synthetic labeling pipeline consists of four steps: caption generation, concept extraction, vocabulary expansion and label assignment. The caption generation process employs off-the-shelf multi-modal LLMs and is straightforward.

For the second step, we prompt LLaMA 2-13B-Chat to extract concept labels for videos from captions generated in Stage 1.The LLM prompt for extracting labels from captions is provided in Listing I. We provide 3 in-context examples and the LLM is prompted to extract concept labels from the fourth video's captions.

As LLMs identify a large number of concepts, including many near-duplicates, a cleanup step is necessary to minimize these issues. We utilize the CLIP text encoder to obtain embeddings for all the identified concepts across the dataset. K-Means clustering is applied to the embeddings to cluster them into groups. For each group, we replace the labels with the most frequently observed concepts

from that group. This works reasonably well, and CLIP text encoder is excellent at detecting near-duplicate visual concepts. A random sample of identified clusters are shown in Table 13.

Finally, we reuse the CLIP text encoder to match labels from the deduplicated vocabulary back to videos, which are represented by their captions. In order to reduce domain shift between captions and the labels, standard CLIP prompt template `"a video of {label}"` is used.

For extracting extra action labels, we use IDEFICS-9b-Instruct model [18] and LLaVA [28] to caption the videos. Both models are based on LLaMA LLMs, with IDEFICS trained using the interleaved image text dataset OBELICS, while LLaVA is trained on a mix of image-caption data and instruction following data created using GPT-3 and image annotations. This stage is followed by LLaMA 2-13b-chat [48] to extract the labels from the captions. OpenAI CLIP B/32 is used to clean up label assignment to videos.

Airlines	'american airlines', 'delta air lines', 'southwest airlines', 'singapore airlines', 'air france', 'emirates (airline)', 'british airways', 'carnival cruise line'
Grilling	'barbequing', 'cooking on campfire', 'grilling', 'barbecue'
Lego	'legoland', 'lego star wars', 'lego minecraft', 'lego duplo', 'the lego group', 'lego friends', 'lego batman: the videogame', 'lego', 'lego batman 3: beyond gotham', 'lego ninjago', 'lego minifigure', 'playmobil', 'lego marvel super heroes', 'lego city', 'lego legends of chima'
Playground	'playing on a playground', 'playground', 'amusement park', 'amusement arcade', 'amusement ride', 'water park', 'ferris wheel'
Video Games	'gears of war (video game)', 'jill valentine', 'gears of war', 'silent hill 2', 'resident evil 2', 'hitman: absolution', 'gears of war 2', 'resident evil 5', 'resident evil 3: nemesis', 'resident evil', 'resident evil (1996 video game)', 'resident evil (2002 video game)'
Water Slide	'water sliding', 'riding water slide', 'water slide'

Table 13: Sampled clusters among concepts identified by the captioning + LLM steps of the label generation pipeline. CLIP Text encoder features were used to cluster the concepts for de-duplication. Cluster names assigned in the left column are only used for illustration.

Our LLM prompt for extracting action labels from video captions.

```
Following is the description of a video. Output a numbered list of verbs
representing visual actions performed in the video. Do not add any explanation.

Video 1 description:
1. A group of people riding motorcycles at night.
2. A motorcycle is lit up with blue lights.
3. A person is riding a bike at night.
4. A motorcycle parked on the street at night.
5. A group of people are gathered in a dimly lit room.
6. A motorcycle parked in a dark room.
7. A motorcycle is parked in a dark room.
8. A person is riding a bike at night.

Verbs Found:
1. riding motorcycle
2. riding bike

Following is the description of a video. Output a numbered list of verbs
representing visual actions performed in the video. Do not add any explanation.

Video 2 description:
1. A man is performing on stage with a band.
2. A group of men are performing on a stage.
3. A man with a microphone is performing on stage.
4. A group of young men performing on stage.
5. A man is singing on a stage with a band.
6. A man is playing a guitar on a stage.
7. A man and a woman are performing on stage.
8. A dark room with a bright light shining on it.

Verbs Found:
1. performing on stage
2. singing on stage
3. playing guitar

Following is the description of a video. Output a numbered list of verbs
representing visual actions performed in the video. Do not add any explanation.


Video 3 description:
1. A person is putting lotion on another person's hand.
2. A person is putting nail polish on another person's nails.
3. A person is putting nail polish on their nails.
4. A person is holding a ball point pen.
5. A person is writing on a piece of paper.
6. A person is holding another person's hand.
7. A person is putting a ring on another person's finger.
8. A black screen with a white frame.

Verbs Found:
1. putting lotion
2. putting nail polish
3. writing
4. putting ring

Following is the description of a video. Output a numbered list of verbs
representing visual actions performed in the video. Do not add any explanation.


Video 4 description:
<output_captions>

Verbs Found:
```

J Baselines

J.1 CLIP + LLM Frozen Baseline

This baseline is an extension of and inspired by prior works [32, 36] that utilize LLMs to prompt CLIP for image classification to our problem of video classification. The LLM is prompted to generate class descriptors utilizing its extensive world knowledge. CLIP can then be used to match these descriptors to video frames to classify them.

The overall process is illustrated in Figure 8a. Unlike the image classification setting it includes a mean pooling operation across frames to get the video level feature. Note that the LLM prompt is designed to elicit output in the form of a list. This simplifies the post-processing of the text output to generate CLIP prompts (Figure 8c). Firstly, the text is split into each item of the list, followed by removal of repetitions (common for this generation of LLMs). Finally we use a standard CLIP prompting template to incorporate both the class label and the descriptor. Sample descriptors for some classes from our downstream datasets are provided in Figure 8b. CLIP + LLM is a reasonably and consistently strong baseline across datasets, as it inherits CLIP's robustness.

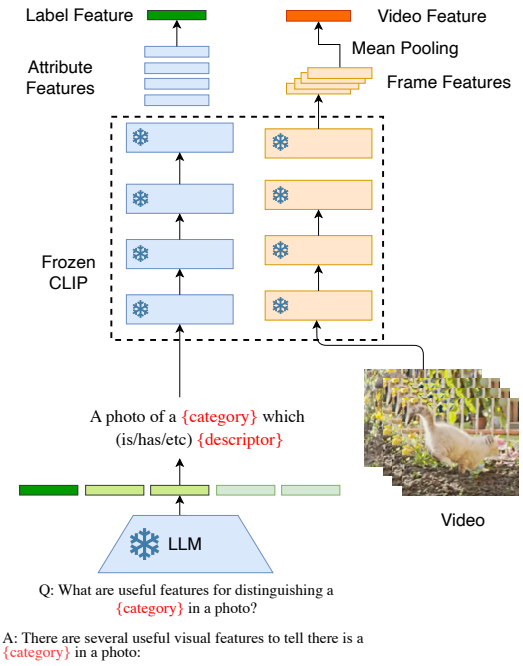
A key limitation of this baseline is the frequency of LLM failures. Different classes of failures such as getting trapped in a repetition loop, generating descriptors which are not visual, and semantic confusion are common. An end-to-end trainable approach could potentially alleviate some of these issues.

J.2 CoOp: Context Optimization

CoOp [61] learns prompts for CLIP's text encoder to adapt it for image classification. This is a parameter efficient adaptation method since it has very few learnable parameters. We extend it to the video setting by utilizing mean pooling across frame in the vision encoder and learnable prompts in the text encoder. (See Figure 9a)

J.3 DualCoOp

DualCoOp [46] refines prompt learning for the multi-label setting, with both positive and negative learnable prompts. A label is matched to a video if the similarity score of the video features with the features for the positive prompts is higher than for the negative prompts. A soft prediction score can be obtained by taking a softmax across the two similarity values. (See Figure 9b)



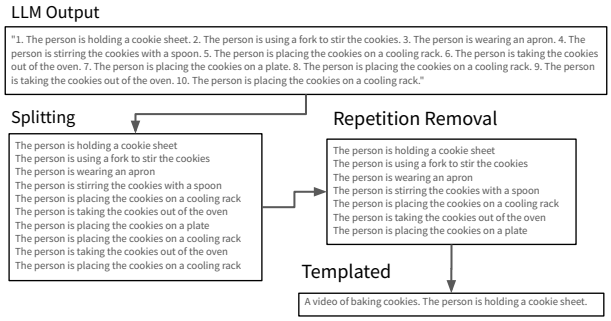
(a) CLIP + LLM Baseline Architecture.

{acorn} **[TAO]**
- The acorn has a hard shell
- The acorn is brown in color
- The acorn is round in shape
- The acorn is attached to a tree
- The acorn is on a tree branch
- The acorn is on a tree trunk

{baking cookies} **[ActivityNet]**
- The person is holding a cookie sheet
- The person is using a fork to stir the cookies
- The person is wearing an apron
- The person is stirring the cookies with a spoon
- The person is placing the cookies on a cooling rack
- The person is taking the cookies out of the oven
- The person is placing the cookies on a plate

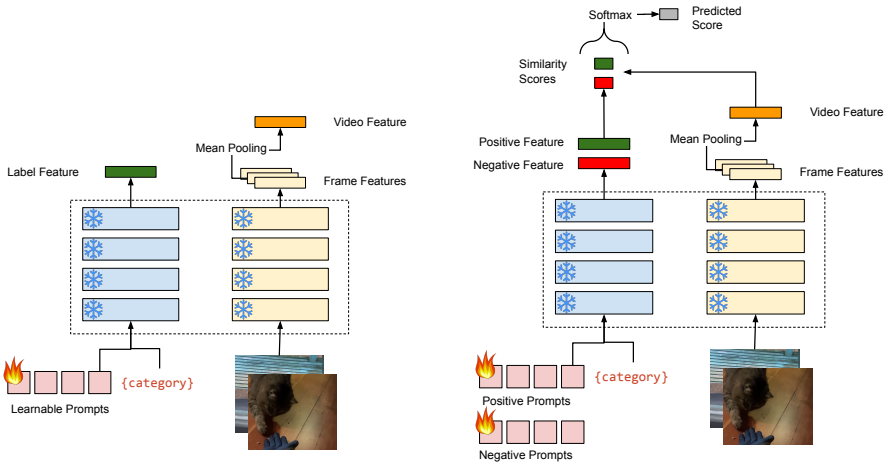
{measure carrots} **[RareAct]**
- The person is holding a tape measure
- They are holding the tape measure with one hand and the carrots with the other hand
- The carrots are on a cutting board
- The person has a knife in their hand
- They are slicing carrots
- They are cutting carrots into pieces
- They are using a grater to grate the carrot pieces
- The carrot pieces are in a bowl
- The bowl has carrots in it
- The bowl is empty

(b) Sample LLM generated Class Descriptors.



(c) Process for cleaning up LLM output to generate CLIP Prompts.

Fig. 8: Here we illustrate the CLIP+LLM baseline discussed in the main paper. **(a)** Architecture consist of frozen CLIP and LLM model. The LLM is prompted to generate class descriptors to assisst CLIP. **(b)** Some sample class descriptors generated by the LLM. **(c)** Process for converting the raw LLM output text to attribute prompts for CLIP. Firstly, the raw text is split into separate list items, then repetitions (which LLMs are prone to) are removed and finally standard CLIP prompting templates are used to combine the class name with the descriptor.



(a) CoOp. **(b)** Dual CoOp.

Fig. 9: Trainable CLIP based baselines without an LLM **(a) CoOp** utilizes learnable prompts on the text encoder side to guide the model towards classification task. **(b) Dual CoOp** is designed for multi-label classification and utilizes learnable positive and negative prompts to generate a probability score for each label.

K Implementation Details

K.1 Datasets Used

For training, we use YouTube-8M and Kinetics datasets. For evaluation we use TAO (Tracking Any Object) dataset for Object Classification and ActivityNet for action classification. We also leverage the RareAct dataset in a novel way by using their noun and verb labels to generate 3 labels for each clip, noun, verb and noun-verb combination. For YouTube-8M, we use the human verified validation set for reporting results. Overall these evaluation datasets cover a wide range of entities and actions, providing a comprehensive evaluation of open vocabulary multi-label video classification capabilities.

Dataset	**# Videos**	**# Classes**	**# Labels/Video**
Training Datasets			
YouTube-8M	2,285,432	2429	2.9
+ Generated Labels	2,285,432	3281	6.7
Kinetics 400	246,245	400	1
+ Generated Labels	246,245	1355	4.5
Test Datasets			
YT-8M Segments Val	42,407	1000	1.05
TAO	655	1230	1.44
ActivityNet	4,593	200	1.01
RareAct	905	214	3.02

Table 14: Details about datasets used for training. For YouTube-8M and Kinetics, we also generate additional labels for training using our synthetic labelling pipeline.

K.2 Training Details

We use Open AI CLIP-B/32 as the Vision-Language model and Google Flan-T5 XL as the promptable LLM.

We use `AdamW` optimizer for training with a base learning rate of 0.00001. Weight decay for newly initialized layers is set to 0.0000001 and 0.0 for CLIP initialized layers. Weight regularization loss weight for STAN's spatial attention layers is set to $\lambda = 0.000001$. Cosine decay learning rate scheduler with warmup is used. Total training length is $30,000$ steps including $2,000$ steps of warmup.

L Qualitative Results

References

1. Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., Vijayanarasimhan, S.: Youtube-8m: A large-scale video classification benchmark (2016) 10
2. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: Proceedings of the International Conference on Machine Learning (ICML) (July 2021) 9
3. Cheng, et al.: VindLU: A recipe for effective video-and-language pretraining. In: CVPR (2023) 4
4. Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S.S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E.H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q.V., Wei, J.: Scaling instruction-finetuned language models (2022) 12
5. Dave, A., Khurana, T., Tokmakov, P., Schmid, C., Ramanan, D.: Tao: A large-scale benchmark for tracking any object. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16. pp. 436–454. Springer (2020) 10
6. Desai, K., Kaul, G., Aysola, Z., Johnson, J.: Redcaps: Web-curated image-text data created by the people, for the people. arXiv preprint arXiv:2111.11431 (2021) 4
7. Fan, L., Krishnan, D., Isola, P., Katabi, D., Tian, Y.: Improving clip training with language rewrites. In: NeurIPS (2023) 4
8. Fang, H., Xiong, P., Xu, L., Chen, Y.: Clip2video: Mastering video-text retrieval via image clip. arXiv preprint arXiv:2106.11097 (2021) 4
9. Gorti, S.K., Vouitsis, N., Ma, J., Golestan, K., Volkovs, M., Garg, A., Yu, G.: X-pool: Cross-modal language-video attention for text-video retrieval. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5006–5015 (2022) 4
10. He, B., Yang, X., Kang, L., Cheng, Z., Zhou, X., Shrivastava, A.: Asm-loc: Action-aware segment modeling for weakly-supervised temporal action localization. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13925–13935 (2022) 1
11. Heilbron, F.C., Niebles, J.C.: Collecting and annotating human activities in web videos. In: Proceedings of International Conference on Multimedia Retrieval. p. 377–384. ICMR '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2578726.2578775, https://doi.org/10.1145/2578726.2578775 10
12. Hu, E.J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=nZeVKeeFYf9 5
13. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 4904–4916. PMLR (18–24 Jul 2021), https://proceedings.mlr.press/v139/jia21b.html 2

14. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning. pp. 4904–4916. PMLR (2021) 4
15. Kaul, P., Xie, W., Zisserman, A.: Multi-modal classifiers for open-vocabulary object detection. In: International Conference on Machine Learning (2023) 4
16. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset (2017) 10
17. Kojima, T., Gu, S.S., Reid, M., Matsuo, Y., Iwasawa, Y.: Large language models are zero-shot reasoners. In: Oh, A.H., Agarwal, A., Belgrave, D., Cho, K. (eds.) Advances in Neural Information Processing Systems (2022), https://openreview.net/forum?id=e2TBb5y0yFf 4
18. Laurençon, H., Saulnier, L., Tronchon, L., Bekman, S., Singh, A., Lozhkov, A., Wang, T., Karamcheti, S., Rush, A.M., Kiela, D., Cord, M., Sanh, V.: Obelics: An open web-scale filtered dataset of interleaved image-text documents (2023) 6
19. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. pp. 3045–3059. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.emnlp-main.243, https://aclanthology.org/2021.emnlp-main.243 5
20. Li, J., Li, D., Xiong, C., Hoi, S.: BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 162, pp. 12888–12900. PMLR (17–23 Jul 2022), https://proceedings.mlr.press/v162/li22n.html 4
21. Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., Hoi, S.C.H.: Align before fuse: Vision and language representation learning with momentum distillation. Advances in neural information processing systems **34**, 9694–9705 (2021) 4
22. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., Chang, K.W., Gao, J.: Grounded language-image pre-training. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10965–10975 (June 2022) 2
23. Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4582–4597. Association for Computational Linguistics, Online (Aug 2021). https://doi.org/10.18653/v1/2021.acl-long.353, https://aclanthology.org/2021.acl-long.353 5
24. Lin, W., Karlinsky, L., Shvetsova, N., Possegger, H., Kozinski, M., Panda, R., Feris, R., Kuehne, H., Bischof, H.: Match, expand and improve: Unsupervised finetuning for zero-shot action recognition with language knowledge. In: ICCV (2023) 4
25. Lin, X., Petroni, F., Bertasius, G., Rohrbach, M., Chang, S.F., Torresani, L.: Learning to recognize procedural activities with distant supervision. arXiv preprint arXiv:2201.10990 (2022) 4
26. Liu, et al.: Mug-STAN: Adapting image-language pretrained models for general video. arXiv:2311.15075 4

27. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: Advances in Neural Information Processing Systems. Curran Associates, Inc. (2023) 4
28. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS (2023) 6
29. Liu, R., Huang, J., Li, G., Feng, J., Wu, X., Li, T.H.: Revisiting temporal modeling for clip-based image-to-video knowledge transferring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6555–6564 (2023) 3, 9, 2
30. Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., Tang, J.: Gpt understands, too. AI Open (2023). https://doi.org/https://doi.org/10.1016/j.aiopen.2023.08.012, https://www.sciencedirect.com/science/article/pii/S2666651023000141 5
31. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. Neurocomputing **508**, 293–304 (2022) 4
32. Menon, S., Vondrick, C.: Visual classification via description from large language models. In: The Eleventh International Conference on Learning Representations (2023), https://openreview.net/forum?id=jlAjNL8z5cs 2, 4, 6, 8
33. Miech, A., Alayrac, J.B., Laptev, I., Sivic, J., Zisserman, A.: Rareact: A video dataset of unusual interactions. arxiv:2008.01018 (2020) 10
34. Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., Wang, X., Zhai, X., Kipf, T., Houlsby, N.: Simple open-vocabulary object detection. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 728–755. Springer Nature Switzerland, Cham (2022) 2
35. Ni, B., Peng, H., Chen, M., Zhang, S., Meng, G., Fu, J., Xiang, S., Ling, H.: Expanding language-image pretrained models for general video recognition. In: European Conference on Computer Vision. pp. 1–18. Springer (2022) 4
36. Pratt, S., Covert, I., Liu, R., Farhadi, A.: 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. pp. 15691–15701 (2023) 2, 4, 8
37. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 2, 4, 12
38. Rasheed, H., khattak, M.U., Maaz, M., Khan, S., Khan, F.S.: Finetuned clip models are efficient video learners. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (2023) 11, 3
39. Rizve, et al.: VidLA: Video-language alignment at scale. In: CVPR (2024) 4
40. Roth, K., Kim, J.M., Koepke, A.S., Vinyals, O., Schmid, C., Akata, Z.: Waffling around for performance: Visual classification with random words and broad concepts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15746–15757 (October 2023) 4
41. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems **35**, 25278–25294 (2022) 4
42. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv preprint arXiv:2111.02114 (2021) 4

43. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565 (2018) 4
44. Shvetsova, N., Kukleva, A., Hong, X., Rupprecht, C., Schiele, B., Kuehne, H.: Howtocaption: Prompting llms to transform video annotations at scale (2023) 4
45. Singh, A., Hu, R., Goswami, V., Couairon, G., Galuba, W., Rohrbach, M., Kiela, D.: Flava: A foundational language and vision alignment model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15638–15650 (2022) 4
46. Sun, X., Hu, P., Saenko, K.: Dualcoop: Fast adaptation to multi-label recognition with limited annotations. Advances in Neural Information Processing Systems **35**, 30569–30582 (2022) 2, 4, 11, 8
47. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: The new data in multimedia research. Commun. ACM **59**(2), 64–73 (jan 2016). https://doi.org/10.1145/2812802, https://doi.org/10.1145/2812802 4
48. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023) 6
49. Wasim, S.T., Naseer, M., Khan, S., Khan, F.S., Shah, M.: Vita-clip: Video and text adaptive clip via multimodal prompting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23034–23044 (2023) 2, 3, 4
50. Weng, Z., Yang, X., Li, A., Wu, Z., Jiang, Y.G.: Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization (2023) 2, 4
51. Weng, Z., Yang, X., Li, A., Wu, Z., Jiang, Y.G.: Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization. In: ICML (2023) 9
52. Wortsman, M., Ilharco, G., Kim, J.W., Li, M., Kornblith, S., Roelofs, R., Lopes, R.G., Hajishirzi, H., Farhadi, A., Namkoong, H., et al.: Robust fine-tuning of zero-shot models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7959–7971 (2022) 4
53. Xu, Z., Zhu, Y., Deng, T., Mittal, A., Chen, Y., Wang, M., Favaro, P., Tighe, J., Modolo, D.: Challenges of zero-shot recognition with vision-language models: Granularity and correctness (2023) 6
54. Xue, H., Sun, Y., Liu, B., Fu, J., Song, R., Li, H., Luo, J.: Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. arXiv preprint arXiv:2209.06430 (2022) 4
55. Yang, Y., Panagopoulou, A., Zhou, S., Jin, D., Callison-Burch, C., Yatskar, M.: Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19187–19197 (June 2023) 4
56. Yao, L., Huang, R., Hou, L., Lu, G., Niu, M., Xu, H., Liang, X., Li, Z., Jiang, X., Xu, C.: Filip: Fine-grained interactive language-image pre-training. arXiv preprint arXiv:2111.07783 (2021) 4
57. Yousaf, A., Naseer, M., Khan, S., Khan, F., Shah, M.: VIDEOPROMPTER: AN ENSEMBLE OF FOUNDATIONAL MODELS FOR ZERO-SHOT VIDEO UNDERSTANDING (2024), https://openreview.net/forum?id=9F0xInGNBF 2

58. Yu, J., Wang, Z., Vasudevan, V., Yeung, L., Seyedhosseini, M., Wu, Y.: Coca: Contrastive captioners are image-text foundation models. arXiv preprint arXiv:2205.01917 (2022) 4
59. Zhang, C., Lu, T., Islam, M.M., Wang, Z., Yu, S., Bansal, M., Bertasius, G.: A simple llm framework for long-range video question-answering. CoRR **abs/2312.17235** (2023), https://doi.org/10.48550/arXiv.2312.17235 2
60. Zhao, Y., Misra, I., Krähenbühl, P., Girdhar, R.: Learning video representations from large language models. In: CVPR (2023) 4
61. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022) 4, 11, 8
62. Zhu, X., Zhang, R., He, B., Guo, Z., Zeng, Z., Qin, Z., Zhang, S., Gao, P.: Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2639–2650 (October 2023) 4