Zero-Shot Compositional Video Learning with Coding Rate Reduction

Supplementary Material

This supplementary material provides detailed descriptions of the experimental setup in the main paper, as well as efficiency analysis and additional qualitative examples.

A. Experimental setup of the Sth-Com dataset

The Sth-Com [30] dataset has 161 primitive temporal (i.e. verb) categories and 248 primitive spatial (i.e. object) categories. These primitive categories are present in both the training and evaluation stages. The action labels are defined by combining these two types of primitive categories, such as (*Opening [something]*, *book*), resulting in a total of 5124 action labels. Throughout the evaluation, these action labels are divided into two groups: ***seen*** classes that are encountered during training and ***unseen*** classes that are never seen during training. For example, as shown in Figure 1 (a), only (*Putting*, *remote*) and (*Taking*, *book*) are used as training data, but the model is required to predict not only these seen classes but also unseen classes (*Putting*, *book*), (*Taking*, *remote*) that are combinations of seen primitive concepts(*remote*, *book*, *Putting*, *Taking*).

To this end, we encode primitive spatial and temporal category labels into fixed-sized vectors. As described in the main paper, we use fastText [2] word embeddings and CLIP [47] text encoder to encode labels for two different experiments following the previous work [30]. Specifically, we initialize the label embeddings with the averaged fasttext word embedding of class names. For the experiment using CLIP, we fed the entire sentence of each class name into the CLIP text encoder with the simple prefix "a " only for object categories. These label embeddings are learnable during training. The predictions for primitive categories are made by logits computed as cosine similarities between the final representation from the model and spatial/temporal label embeddings with proper normalization.

B. Experimental setup of the CATER dataset

Similar to the setting in the Sth-Com dataset, the CATER [11] dataset has 4 primitive temporal categories (*rotate*, *pick-place*, *slide*, *contain*) and 5 primitive spatial categories (*cube*, *sphere*, *cylinder*, *cone*, *snitch*). The action labels are defined by combining these two types of primitive categories, which are present both in training and evaluation. This results in 14 atomic action labels excluding the physically infeasible actions such as (*sphere*, *contain*). These action labels are divided into ***seen*** and ***unseen*** classes throughout the evaluation, similar to the experiments on the Sth-Com. For example, as shown in Figure 1 (b), suppose the model only encounters atomic action classes of (*sphere*, *slide*) and (*cube*, *pick-place*). Then, it is required to predict not only these two seen classes but also unseen classes of (*cube*, *slide*) and (*sphere*, *pick-place*) by recombining

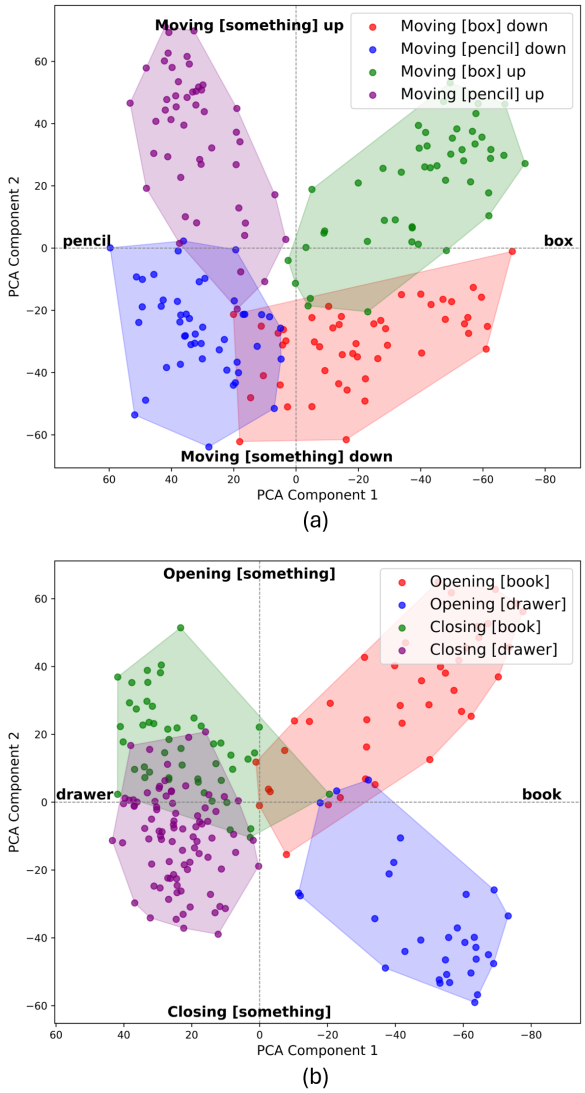

Figure 5. Additional PCA results of the resampled representations learned by our method for two representative scenarios from the Sth-Com [30] dataset. In each scenario, we visualize the first two principal components of the resampled representations. These representations are derived from videos belonging to 4 possible action labels that are the combination of pre-defined two different categories for each spatial and temporal attribute. It can be observed that the principal component axes of learned representations are closely aligned with the semantic axes of attributes.

Table 7. GFLOPs comparison for a single video input.

Method	# Frames			
	8	16	32	64
w/o ΔR	17.59	30.64	56.74	108.94
w/ ΔR	18.57 (+5.57%)	31.62 (+3.19%)	57.72 (+1.72%)	109.92 (+0.80%)

learned primitive concepts (*sphere*, *cube*, *slide*, *pick-place*).

To evaluate the proposed method for zero-shot compositional action recognition on the CATER dataset, we use a recently proposed novel data split [58] different from the original split. In detail, atomic action labels in the CATER dataset are divided into three disjoint groups, let L_A, L_B, and L_C. Then, action labels belonging to $L_A \cup L_C$ are categorized as ***seen*** classes, and action labels belonging to L_B are categorized as ***unseen*** classes. In other words, the model was trained on a subset of videos having labels belonging to $L_A \cup L_C$, and it was evaluated for the remaining videos having labels from $L_A \cup L_C$, which were seen during training, as well as videos having labels in L_B, which were unseen during training. To implement this setting, we use the embedding lookup to encode primitive spatial and temporal categories, similar to the experiments on the Sth-Com, excluding that label embeddings are randomly initialized to make a fair comparison with baseline methods reported in [58].

C. Experimental setup with VLMs

As described in the main paper, we compare the proposed method with the most recent large-scale vision-language models (VLMs), InternVL2-8B [7] and Qwen2-VL-8B [50], for a more rigorous analysis. Specifically, the primitive label (object, motion) candidates along with the corresponding video are provided as input, and we fine-tune them with LoRA [17] adapters to predict composite action labels. The illustration of the input sequence example is provided in Figure 6. The experiment on the Sth-Com was infeasible due to the excess of context length by a large number of primitive spatial and temporal categories (248 and 161, respectively). For evaluation, we take a post-processing step where raw predictions are adjusted using a cosine similarity-based approximation to ensure the answer output is properly aligned with the label taxonomy in the dataset.

D. Efficiency

The proposed method circumvents the computational complexity by compressing the input feature into a fixed-size set of latent queries in the resampler module, and each layer performs a single rate reduction optimization step. Therefore, the total complexity of the model forward pass linearly scales with the length of the input feature map and the number of layers. To empirically validate this analysis, we provide GFLOPs of the resampler module by varying the number of frames in the input video. Table 7 shows the GFLOPs of the forward pass of the resampler module given a single video input having a resolution of 224x224 and a patch size of 16, with a varying number of frames. The results validate that complexity scales linearly with the frame count, and the rate reduction optimization steps add only a negligible amount of overhead.

Input sequence used to train InternVL2 and Qwen2-VL

<|im_start|> user
<|vision_start|> video.avi <|vision_end|> You are an AI agent tasked with analyzing a video and identifying the actions present in it.
Each action is a combination of an object and the motion it performs.
The possible objects and motions in the video are as follows:
Objects: sphere, cylinder, snitch, cone, cube
Motions: pick place, slide, rotate, contain
For each action present in the video, report it as a combination of the object and motion in the following format:
e.g., sphere_slide, cube_rotate, cone_slide
List the final actions present in the video, separated by commas.
<|im_end|>

<|im_start|> assistant
Actions exist in the video are : cone_contain, cone_pick_place, cube_pick_place, cone_pick_place, cone_pick_place, cylinder_rotate
<|im_end|>

Figure 6. Illustration of the input sequence example used to train VLMs for zero-shot compositional action recognition on the CATER [11] dataset.

E. Additional PCA examples

Figure 5 shows additional PCA results similar to Figure 4 in the main paper. In this experiment, we explore the characteristics of subspaces of learned representations to validate whether the proposed method learns desired representations. To this end, we first define two distinct categories for spatial and temporal attributes, focusing on the challenging case of opposite temporal attributes. For example, as seen in Figure 5 (a), we select spatial categories (*box*, *pencil*) and opposing temporal categories (*Moving [something] up*, *Moving [something] down*). Then, we aggregate the learned representations from our method for videos having action labels (*Moving [something] up*, *book*), (*Moving [something] up*, *pencil*), (*Moving [something] down*, *book*), (*Moving [something] down*, *pencil*), which are four possible combinations of predefined spatial and temporal categories. The PCA results of this representation are shown in Figure 5 (a), where the first two principal components of the learned representations are both closely aligned with spatial and temporal semantic axes. A similar example with a different combination of spatial and temporal categories can be found in Figure 5 (b).

Figure 7. The additional examples of cross-attention map visualization in the last layer of the resampler module. Example videos are selected in the Sth-Com [30] dataset, having six different action labels: (a) *Pushing lipstick from right to left*, (b) *candy falling like a rock*, (c) *Pulling tape from left to right*, (d) *Tilting lid with a clothespin on it until it falls off*, (e) *Moving cup across a surface until it falls down*, (f) *Dropping slipper onto floor*. For each video, we select a representative head and visualize its attention map on the input frames across time. It can be observed that each head captures the informative object and its temporal variations, even trained without dense labels of object positions such as bounding boxes.

F. Additional attention map examples

Figure 7 shows additional examples of cross-attention map visualization in the last layer of the proposed resampler module, similar to Figure 3 in the main paper. We emphasize again that the proposed method captures the informative object within a video and its temporal variations very clearly even though it is trained without dense labels giving the precise position of the object, such as bounding boxes.