

# Supplementary Materials: Generating Action-conditioned Prompts for Open-vocabulary Video Action Recognition

Anonymous Authors

This supplementary material offers extensive additional details and more qualitative and quantitative analysis complementing the main paper. The content is organized as follows:

- Prompts for LLMs and Responses (Appendix 1)
- Ablation on LLM-prompts and attributes (Appendix 2)
- Additional qualitative visualization (Appendix 3)
- More details of the training objectives and procedures (Appendix 4)
- More implementation details (Appendix 5)
- More details of experimental settings (Appendix 6).
- More details of datasets (Appendix 7).

## 1 PROMPTS FOR LLMs AND RESPONSES

Drawing inspiration from LLMs’ chain-of-thought [7, 12], we adopt a Action-Centric generation strategy to generate multi- attribute Action-conditioned Prompts. Initially, we prompt the LLM to enumerate a list of attributes crucial for distinguishing actions. Subsequently, the second prompt directs the LLM to formulate descriptions utilizing the attributes identified in the first step, while specifying the desired output format. We prefer a JSON dictionary format for the output, comprising a list of sentences, each linked to a specific attribute. This format is not only straightforward to utilize but also maintains the granularity of attribute-level information, thereby promoting a richness in attributes.

### 1.1 Which attributes are critical for action recognition?

Example of the first prompt for determining attributes:

*Q: I am currently utilizing CLIP for video action classification tasks. I want to create action class attributes for a zero-shot video action recognition algorithm to classify different actions. List some attributes that take into account the scene of the action’s occurrence, information about the actor performing the action, and details of the actor’s bodily movements, which can contribute to a class description that aids in distinguishing actions.*

GPT-4 response:

*A: For creating effective action class attributes for zero-shot video action recognition using a tool like CLIP, it’s crucial to integrate aspects that cover the scene, the actor, and the nature of the movements. Here are some attributes that can be considered for each category: ...*

In Table 1, we display the list of attributes generated by GPT. After removing duplicates and unimportant content, we finalized a set of 12 core attributes. These are distributed across the three components and are illustrated in Figure 3 in the main paper.

### 1.2 Describe the action about critical attributes

We then construct a set of LLM-prompts. Their purpose is to inquire specifically about certain actions in relation to identified attributes, aiming to generate standardized, knowledge-enriched descriptive sentences. Examples of LLM-prompts are illustrated as follows:

*LLM-prompt1: Describe the following actions by adding one sentence about its {attribute} for {action}.*

*LLM-prompt2: Visually describe {action} about its {attribute} in one sentence.*

*LLM-prompt3: What are the primary characteristics of {action} interm of its {attribute}, explain in one sentence.*

*LLM-prompt4: Explain how {action} typically manifests in terms of its {attribute}, providing a descriptive sentence.*

*LLM-prompt5: Analyze {action} in the context of its {attribute}, and describe it in one sentence.*

Then, for the selected LLM-prompt, we generate a suite of 12 distinct action-conditioned prompts, ensuring that every action is matched with tailored descriptive phrases, as depicted in Figure 3 in the main paper. Additionally, we instructed GPT to output the results in a JSON format, where each key is an attribute and the corresponding value is the descriptive sentence for that attribute.

Using *Making Sushi*, *Drumming* and **LLM-prompt1** as an example, GPT’s output response is as follows:

*“Making Sushi”: { “Scene Type”: “This action typically occurs in a kitchen or sushi bar.”, “Scene Elements”: “The scene includes a countertop, a bamboo sushi mat, rice, nori (seaweed sheets), and various fillings like fish, vegetables, and condiments.”, “Scene Conditions”: “The environment is usually controlled, indoor settings.”, “The usage of props in the action”: “Props used include a bamboo sushi mat, a sharp knife, and a rice cooker.”, “Number of Actors”: “This action can be performed by a single individual or with others, especially in a professional setting.”, “Clothing”: “Chefs’ attire or aprons may be worn.”, “Actor Identity”: “Anyone who enjoys cooking or is employed as a sushi chef can perform this action.”, “Facial Expression”: “Expressions can range from focused to satisfied, especially when creating or enjoying a beautiful sushi roll.”, “Body move Speed”: “The action is performed at a moderate to slow pace, allowing for precision in rolling the sushi.”, “Body Part Movement”: “Hand and arm movements are used to spread the rice, place the fillings, and roll the sushi using the bamboo mat.”, “Body Posture”: “Individuals are typically standing while performing this action.”, “Body Position”: “The body remains on the ground during this action.” },*

Type	Attributes	Explanation
Scene-Related	Scene Type *	Soccer field, Kitchen, Street, Gymnasium, etc.
	Scene Elements *	Soccer ball, Goalpost, Stove, Traffic sign, etc.
	Scene Conditions *	Sunny, Rainy, Indoor, Outdoor, etc.
	Prop Usage *	Soccer ball, Knife, Cookware, etc.
	Weather Conditions	Sunny, cloudy, rainy, snowy, foggy, etc.
	Human Crowds	Busy streets, empty spaces, group gatherings, etc.
	Specific Locations	Parks, offices, classrooms, industrial areas, etc.
	Terrain Type	Flat ground, hilly area, uneven surfaces, water bodies, etc.
	Cultural Context	Specific to a region or community.
	Color and Texture	Bright, dark, colorful, monochrome environments, etc.
Actor-Related	Number of Actors *	Single, Double, Multiple.
	Clothing *	Sportswear, Chef's uniform, Police uniform, etc.
	Actor Identity *	Athlete, Chef, Policeman, etc.
	Facial Expression *	Happy, Sad, Angry, Surprised, etc.
	Age Group	Children, teenagers, adults, elderly.
	Clothing Style	Formal, casual, athletic, traditional.
	Emotional State	Stressed, calm, excited, bored.
	Hairstyle and Accessories	Short, long hair, hats, glasses.
	Visible Health Conditions	Signs of fatigue, injury, robust health.
	Ethnicity or Cultural Background	Diverse cultural representations.
Body-Related	Body Move Speed *	Fast, Medium, Slow, etc.
	Body Part Movement *	Hand, Leg, Head, etc.
	Body Posture *	Standing, Sitting, Lying, Bending, etc.
	Body Position *	In contact with ground, Off the ground, etc.
	Purpose of Body Movement	Functional, expressive, recreational, competitive, etc.
	Changes in Posture	Standing, sitting, lying, bending, etc.
	Movement Complexity	Simple, complex, repetitive, unique, etc.
	Body Coordination Level	Coordinated, uncoordinated, synchronized, etc.
	Body Movement Style	Graceful, abrupt, fluid, stiff, etc.
	Body Rhythm and Timing	Regular, irregular, rhythmic, sporadic, etc.

**Table 1: List of attributes and their corresponding explanations as provided by GPT responses. Attributes marked with an asterisk (\*) are those that were ultimately selected.**

*"Drumming": { "Scene Type": "The action typically occurs on a stage, in a music studio, or in a practice room.", "Scene Elements": "The scene contains a drum set, drumsticks, and possibly other musical instruments and musicians.", "Scene Conditions": "The area is well-lit and acoustically suitable for playing music.", "The usage of props in the action": "Drumsticks and a drum set are the main props used in this action.", "Number of Actors": "Usually, one person plays the drums, although other musicians may be present.", "Clothing": "The drummer wears casual or performance attire, depending on the setting.", "Actor Identity": "The actor is a drummer, possibly*

*part of a band or ensemble.", "Facial Expression": "The drummer may exhibit focus, enjoyment, and rhythm as they play.", "Body move Speed": "The action varies in speed, with fast, rhythmic drumming or slower, deliberate strikes.", "Body Part Movement": "The drummer's arms move rapidly to strike the drums, while the feet operate the bass drum and hi-hat pedals.", "Body Posture": "The body posture is seated with a straight back, and arms and legs in motion.", "Body Position": "The drummer remains seated on a stool during the action." }*

**The complete prompts will be made publicly available promptly after the paper is accepted.**

**Table 2: The impact of different LLM-prompts.**

Method	HMDB-51	UCF-101	K-600
LLM-prompt1	54.7	<b>81.1</b>	<b>72.4</b>
LLM-prompt2	54.3	80.8	72.2
LLM-prompt3	54.5	81.0	72.1
LLM-prompt4	<b>54.8</b>	80.5	72.3
LLM-prompt5	54.1	80.7	71.8
LLM-prompt num 2	55.1	81.9	73.3
LLM-prompt num 3	<b>55.4</b>	82.4	<b>73.4</b>
LLM-prompt num 4	55.3	<b>82.7</b>	<b>73.4</b>
LLM-prompt num 5	55.2	82.6	73.7

**Table 3: The impact of different attributes.**

Method	HMDB-51	UCF-101	K-600
Scene	52.5	82.0	72.5
Actor	54.1	81.4	72.9
Body	54.8	81.3	72.5
Scene+Actor	54.2	82.3	73.1
Scene+Body	54.5	82.1	73.1
Actor+Body	55.1	82.1	73.3
Scene+Actor+Body	<b>55.4</b>	<b>82.4</b>	<b>73.4</b>

## 2 ABLATION ON LLM-PROMPTS AND ATTRIBUTES.

We present additional ablation studies including the impact of different LLM-prompts and attributes.

### 2.1 The impact of different LLM-prompts

Table 2 showcases the impact of different LLM-prompts on performance. The upper section presents results using individual LLM-prompt, while the lower section shows the collective performance when selecting the best-performing set of {num} prompts. It is observable that variations in LLM-prompts have a minimal effect on performance (less than 0.6%), underscoring their robustness compared to manually designed prompts fed into CLIP. Furthermore, while integrating multiple LLM-prompts can enhance performance, the gains become marginal as the number increases. Considering the trade-off between efficiency and performance, we default to employing a set of three LLM-prompts, including *LLM-prompt1*, *LLM-prompt3*, and *LLM-prompt4*.

### 2.2 The impact of different attributes

Table 3 illustrates the performance impact of prompts associated with different attributes across various datasets. It is evident that prompts tied to specific attributes can enhance model performance. However, the degree of improvement attributed to each category varies depending on the dataset. For instance, prompts related to "Body" show a greater benefit for HMDB-51, while "Actor" prompts

yield more substantial gains for UCF-101 and Kinetics-600, indicating a divergence in focal points among datasets. Overall, prompts that amalgamate all three aspects, *i.e.*, Scene, Actor, and Body, achieve a comprehensive performance boost across datasets, which mitigates dataset biases. This underlines the necessity of our Action-Centric generation approach in producing multi-attribute prompts and its effectiveness in addressing the varying scenarios of different datasets.

## 3 ADDITIONAL QUALITATIVE VISUALIZATION

We provide a more detailed visualization of the frame-to-prompt correspondence in Figure 1.

## 4 MORE DETAILS OF THE TRAINING OBJECTIVES AND PROCEDURES

Given a video  $V \in \mathbb{R}^{T \times H \times W \times 3}$  with  $T$  frames and a set of prompts  $C$ , where  $V$  and  $C$  are sampled from a set of videos  $\mathcal{V}$  and a collection of action category  $C$  respectively, we feed the  $T$  frames into the video encoder  $f_{\theta_v}$  and the text  $C$  into the text encoder  $f_{\theta_t}$  to obtain a video representation  $v \in \mathbb{R}^{n_v \times d}$  and text embeddings  $c \in \mathbb{R}^{n_t \times d}$  correspondingly.

For a batch of videos, the similarity  $\text{sim}(\cdot)$ , between all the video representation  $v$  and the corresponding text embeddings  $c'$  is maximized to fine-tune the CLIP model via cross-entropy (CE) objective with a temperature parameter  $\tau$ ,

$$\mathcal{L} = - \sum_{v \sim \mathcal{V}} \log \frac{\exp(\text{sim}(v, c')/\tau)}{\sum_{c \sim C} \exp(\text{sim}(v, c)/\tau)},$$

where  $c'$  represents the ground truth action-conditioned prompts corresponding to video  $v$ . We employ Equation (??) in the main paper to compute the fine-grained similarity between prompts and each video.

We also provide a PyTorch-style pseudocode for the *multi-modal action knowledge alignment* mechanism in Algorithm 1 to aid in understanding the entire alignment procedure.

## 5 MORE IMPLEMENTATION DETAILS

Our adaptation of the CLIP model follows [9], with tailored modifications to the prompts and fine-grained similarity function used. We preprocess all frames to a uniform spatial dimension of  $224 \times 224$  pixels. Optimization is carried out using an AdamW optimizer with a weight decay set at 0.001. Adaptations in epochs, batch size, and learning rate are made to suit varying experimental conditions, as outlined subsequently. For the *zero-shot setting*, CLIP is trained on the Kinetics-400 dataset for 10 epochs, utilizing a batch size of 256 and a learning rate of  $8e-6$ . In both the *base-to-novel generalization* and *few-shot settings*, training proceeds in a few-shot manner with a batch size of 64 and a learning rate of  $2e-6$ . Under the *fully-supervised setting*, we extend CLIP's training on Kinetics-400 to 30 epochs, with an increased batch size of 256 and a learning rate of  $22e-6$ . These experiments were performed on a computing cluster equipped with 8 A100 GPUs.

As our method adopts action-conditioned prompts, it diverges from the ViFI approach which utilizes learnable prompting methods. For experiments transitioning from utilizing parameters pre-trained



**Figure 1: Illustrative heatmap of the association strengths between 8 video frames and a set of 12 prompts exemplified for the actions “Making Sushi” and “Drumming”, as detailed in Appendix 1.2. The intensity of the color corresponds to the association strength, calculated using the CLIP match score. The heatmap reveals that different frames are associated with prompts from varying attributes, providing a clear pathway to understanding how the model discerns actions through both visual and textual cues.**

on Kinetics-400 to the base-to-novel and few-shot scenarios, we fine-tune the pre-trained CLIP model directly, with a batch size of 64 and a learning rate of  $2e-6$ . Empirical evidence from our experiments corroborates the effectiveness of this approach, as shown in Table 2 and 3 in the main paper.

## 6 MORE DETAILS OF EXPERIMENTAL SETTINGS

We align with previous methods [4, 8, 9, 11] for various settings including zero-shot, base-to-novel, few-shot, and fully-supervised. Specifically, we utilize 8 frames and employ multi-view inference incorporating 2 spatial crops and 2 temporal views. In the fully supervised setting, our approach extends to using 16 frames, combined with multi-view inference featuring 4 spatial crops and 3 temporal views, consistent with compared methods. Each sampled frame is spatially scaled on the shorter side to 256, with a center crop of 224.

**Zero-shot setting:** In the zero-shot setting, models trained on the Kinetics-400 dataset undergo testing on three distinct datasets: HMDB-51, UCF-101, and Kinetics-600. For HMDB-51 and UCF-101, performance is assessed across the three standard validation splits, with the top-1 average accuracy being reported. Regarding Kinetics-600, following the methodology of [2], the evaluation focuses on the 220 categories that do not overlap with those in Kinetics-400. Here, we also document top-1 average accuracies derived from three randomly generated splits, each inclusive of 160 categories.

This assessment utilizes a multi-view strategy, encompassing 2 different spatial crops and 2 temporal clips, amounting to a total of 32 frames.

**Base-to-novel setting:** Following [9], we employ a *base-to-novel generalization* setting for extensive analysis on the generalization ability of various approaches. In this setting, models undergo initial training on a set of ‘base’ (seen) classes using a few-shot approach and are then evaluated on a set of ‘novel’ (unseen) classes. Our analysis spans four datasets: Kinetics-400, HMDB-51, UCF-101, and SSv2 as [9]. For each, we employ three training splits, with 16 shots per action category, selected at random. Categories are divided into two equal groups: the more frequently occurring actions serve as ‘base’ classes, while the less common ones are designated as ‘novel’ classes. Evaluations are performed on the respective validation splits, with HMDB-51 and UCF-101 limited to their first split, and full validation splits used for Kinetics and SSv2. The setting also follows a multi-view strategy, integrating two spatial crops and two temporal clips.

**Few-shot setting:** In the few-shot scenario, we establish a general K-shot configuration by randomly selecting K examples from each category for training purposes. Concretely, for the datasets HMDB-51, UCF-101, and SSv2, we utilize 2, 4, 8, and 16 shots. Performance evaluations for HMDB-51 and UCF-101 are conducted using their first validation split, while for SSv2, the entire validation split is used. This setting also employs a multi-view strategy, integrating two spatial crops and two temporal clips.



---

**Algorithm 1** PyTorch-style pseudocode for multi-modal action knowledge alignment mechanism.
 

---

```

465 # e_c: Action prompts embeddings
466 # e_v: videos embeddings
467 # f_t: text encoder network
468 # f_v: video encoder network
469 # B: batch size
470 # D: dimensionality of the embeddings
471 # K: number of categories
472 # N_v: number of frames
473 # N_t: number of prompts for each action
474
475 def action_knowledge_alignment(C, V):
476     # compute embeddings
477     e_c = f_t(C) # KxN_txD
478     e_v = f_v(V) # BxN_vxD
479
480     # normalize representation
481     e_c = e_c / e_c.norm(dim=-1, keepdim=True)
482     e_v = e_v / e_v.norm(dim=-1, keepdim=True)
483
484     # fine-grained relevancy between prompts and frames
485     logits = torch.einsum('bvd,ktd->bktv', [e_v, e_c])
486     # BxKxN_txN_v
487
488     t2v_logits, t2v_max_idx = logits.max(dim=-1)
489     # BxKxN_txN_v -> BxKxN_t
490     t2v_logits = t2v_logits.mean(dim=-1)
491     # BxKxN_t -> BxK
492     v2t_logits, v2t_max_idx = logits.max(dim=-2)
493     # BxKxN_txN_v -> BxKxN_v
494     v2t_logits = v2t_logits.mean(dim=-1)
495     # BxKxN_v -> BxK
496
497     alignment_logits = (t2v_logits + v2t_logits) / 2.0
498     # BxK
499
500     return alignment_logits
501

```

---

**Fully-supervised setting:** In the fully-supervised setting, models trained on the Kinetics-400 dataset are assessed against its entire validation set. We conduct evaluations using 16 frames and employ a multi-view inference approach, which includes three distinct spatial crops and four temporal segments.

## 7 MORE DETAILS OF DATASET

We conduct our analysis on five established action recognition benchmarks: Kinetics-400 [5] and Kinetics-600 [1], HMDB-51 [6], UCF-101 [10] and Something-Something v2 (SSv2) [3].

**Kinetics-400 and Kinetics-600:** The Kinetics-400 and Kinetics-600 datasets are comprehensive collections designed for human action recognition, containing approximately 240k training and 20k validation videos across 400 action classes, and around 410k training and 29k validation videos covering 600 action classes, respectively. Originating from diverse YouTube videos, each clip is roughly 10 seconds in length, capturing a concise action moment. Kinetics-600 builds upon the foundation set by Kinetics-400, introducing an additional 220 action categories that enrich the dataset, particularly for evaluating zero-shot learning capabilities. While these datasets offer a wide variety in content, it's noteworthy that there is a tendency towards spatial appearance biases. These extensive

collections present an opportunity for models to demonstrate their proficiency in recognizing a broad spectrum of human activities.

**HMDB-51:** The HMDB-51 dataset comprises 6,849 video clips distributed across 51 distinct action categories, ensuring a minimum of 101 clips per category. This dataset has been amassed from various realistic sources and is designed for a balanced evaluation. Officially, it offers three different training/testing splits. To maintain uniformity across categories, each split is configured to include 70 training and 30 test samples per category, while leaving 1,746 videos as 'unused' to preserve sample balance. This structure of training and testing allows for a consistent and fair assessment of the model's performance across the full spectrum of actions.

**UCF-101:** The UCF-101 dataset is a benchmark for human action recognition featuring 13,320 video clips sourced from YouTube, spanning 101 action categories. These categories encompass a broad range of actions including human-object interaction, body motion, human-human interaction, playing musical instruments, and various sports. Each video is a succinct representation of an action, averaging 7.21 seconds in length, derived from realistic scenarios. For evaluation consistency, the dataset is divided into three standard splits, with the official split allocating 9,537 videos for training and 3,783 for testing.

**Something-Something v2 (SSv2):** The SSv2 dataset is a comprehensive video action recognition benchmark that specifically emphasizes temporal modeling. It features 220,487 videos across 174 action categories, capturing humans interacting with everyday objects. The actions depicted in SSv2 are finely detailed, focusing on nuanced activities such as covering or uncovering objects, thereby showcasing a dataset with a strong temporal bias distinct from other datasets like K400. The videos range from 2 to 6 seconds in length, highlighting the rich temporal details over static scenes. The standard dataset split includes 168,913 training videos and 24,777 validation videos. We evaluate and report the top-1 accuracy using the validation split. SSv2 uniquely prioritizes dynamic information in videos over static scene contexts, presenting a challenging environment for models to accurately capture and interpret temporal action dynamics.

## REFERENCES

- [1] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. 2018. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340* (2018).
- [2] Shizhe Chen and Dong Huang. 2021. Elaborative Rehearsal for Zero-shot Action Recognition.
- [3] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. 2017. The "something something" video database for learning and evaluating visual common sense.
- [4] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. 2022. Prompting visual-language models for efficient video understanding.
- [5] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* (2017).
- [6] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. 2011. HMDB: a large video database for human motion recognition.
- [7] Mayug Maniparambil, Chris Vorster, Derek Molloy, Noel Murphy, Kevin McGuinness, and Noel E O'Connor. 2023. Enhancing clip with gpt-4: Harnessing visual descriptions as prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 262–271.

- [8] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. 2022. Expanding language-image pretrained models for general video recognition.
- [9] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. 2023. Fine-tuned clip models are efficient video learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6545–6554.
- [10] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* (2012).
- [11] Mengmeng Wang, Jiazheng Xing, and Yong Liu. 2021. ActionCLIP: A New Paradigm for Video Action Recognition. *arXiv preprint arXiv:2109.08472* (2021).
- [12] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638

639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696