PCBEAR: Pose Concept Bottleneck for Explainable Action Recognition

Supplementary Material

In this supplementary material, we provide implementation/metrics/dataset details, and additional experimental results to complement the main paper. We organize the supplementary material as follows:

1. Complete implementation details
2. Evaluation metric details
3. Dataset details

1. Details of datasets.

KTH-5 The KTH [5] dataset consists of 25 actors, each performing six actions across four different environments. The originally proposed splits ensure that videos of a single actor remain within one set, as the dataset was not designed for identity recognition.

Penn Action The Penn Action [9] dataset includes 2,326 video sequences covering 15 different actions with detailed human joint annotations. Each frame is stored in RGB format with a resolution of up to 640×480 pixels. The dataset provides 2D joint locations, visibility annotations, bounding boxes, coarse viewpoint labels, and train/test splits, making it well-suited for action recognition and pose estimation tasks.

HAA49 The HAA500 [1] dataset consists of 500 fine-grained action classes with over 591K labeled frames, focusing on precise human-centric atomic actions. Unlike coarse-grained datasets, it ensures clear class distinctions (e.g., "Baseball Pitching" vs. "Free Throw in Basketball") and maintains high pose detectability. Carefully curated to eliminate irrelevant motions and label noise, HAA500 is well-suited for detailed action recognition and human pose analysis. The HAA49 dataset is a subset of HAA500, consisting of 49 action classes that focus on fine-grained, human-centric actions. We use the following classes from the HAA49 dataset for sampling: `yoga_bridge`, `yoga_cat`, `yoga_dancer`, `yoga_firefly`, `yoga_fish`, `yoga_gate`, `yoga_locust`, `yoga_lotus`, `yoga_pigeon`, `yoga_tree`, `yoga_triangle`, `yoga_updog`, `weightlifting_hang`, `weightlifting_overhead`, `weightlifting_stand`, `volleyball_overhand`, `volleyball_pass`, `volleyball_set`, `volleyball_underhand`, `tennis_serve`, `taekwondo_middle_block`, `sprint_start`, `soccer_shoot`, `soccer_throw`, `situp`, `pushup`, `punching_sandbag`, `pull_ups`, `one_arm_push_up`, `leg_hold_back`, `leg_hold_front`, `leg_split`, `high_knees`, `handstand`, `gym_squat`, `gym_run`, `gym_ride`, `gym_plank`, `gym_pull`, `gym_push`, `gym_lunges`, `gym_lift`, `golf_swing`, `dips`, `bench_dip`, `baseball_swing`, `backflip`, `arm_wave`, and `jumping_jack`. These carefully curated classes in HAA49 enable detailed analysis of pose-based actions and movements, making it an excellent resource for video action recognition tasks focused on human poses and dynamic motion.

2. Details of implementations.

We implement our framework using PyTorch, built upon the Label-Free CBM [4] code base. The framework utilizes a Nvidia RTX 3090 GPU for training and evaluation. For feature extraction, we use VideoMAE-B/16 [7], pretrained on Kinetics-400 [2]. For pose estimation, we employ the off-the-shelf ViTPose-B [8] model, pretrained on COCO [3], and leverage 17 keypoints from COCO for pose extraction. To create the concept set for the textual concept baseline, as shown in Figure 1 and Table 1 of the main paper, we follow the Label-Free-CBM [4] method for selecting and defining concepts. The Label-Free-CBM provides the code base for concept set processor, which we use to generate concepts tailored to each dataset. To match the number of concepts with the number of classes, we remove concepts that are similar to class names. We train the concept layer with a batch size of 256 for 1,000 steps. The sparse linear layer is optimized with a batch size of 512, a learning rate of 1e-3, $\alpha = 0.99$, and a step size of 0.05 for controlled updates. For input clips, we set the frame length $l = 16$ frames per clip, and each pose is extracted using the 17 keypoints from the MS COCO [3] dataset, with $J = 17$ keypoints for each frame. These configurations ensure that the model efficiently learns both spatial and temporal dynamics, providing high-quality representations for action recognition while maintaining interpretability and robustness.

3. Metric.

To adapt the Concept Utilization Efficiency (CUE) metric from Res-CBM [6] for our framework, we refer to the original formulation:

$$CUE = \frac{10000 \times \text{Acc}}{N \times \bar{L}},$$

where Acc represents the classification accuracy, N is the number of concepts, and $\bar{L}$ is the average length of the concepts.

In Res-CBM, the multiplication by 10,000 is used to scale the metric, allowing it to measure the improvement in classification accuracy considering both the length of the concepts and the total number of concepts. Larger values of CUE indicate higher efficiency in utilizing concepts. For our case, since we are working with pose-based concepts derived from skeleton sequences, we cannot define concept length in the same way as textual concepts in Res-CBM. Therefore, we set $\bar{L} = 1$, as each pose concept corresponds to a fixed-length representation (such as a specific pose or sequence of poses) rather than a variable-length textual description. Additionally, to avoid an overly large scale due to multiplying by 10,000, we use a more manageable scaling factor of 100. This adjustment ensures the CUE score remains interpretable while still reflecting the efficiency of concept utilization. Thus, our modified CUE metric becomes:

$$CUE^* = \frac{100 \times \text{Acc}}{N}.$$

This formula reflects the efficiency of concept utilization in our framework, where Acc is the top-1 accuracy, and N is the number of concepts. This modification enables us to assess the effectiveness of pose-based concepts in video action recognition tasks, with a focus on maintaining simplicity and clarity in the calculation of CUE.

Additionally, to evaluate the effectiveness of our clustering results, we employ the Normalized Mutual Information (NMI) score, which quantifies the alignment between the learned concept clusters and ground truth labels. It computes how much information is shared between the two distributions: the learned clusters and the true labels. A higher NMI score indicates that the clusters are more consistent with the true class labels, reflecting a more accurate clustering of the concepts. NMI is normalized to ensure that the score is independent of the number of clusters, making it a robust and interpretable metric for evaluating the quality of the concept clusters.

References

[1] Jihoon Chung, Cheng-hsin Wuu, Hsuan-ru Yang, Yu-Wing Tai, and Chi-Keung Tang. Haa500: Human-centric atomic action dataset with curated videos. In *ICCV*, 2021. 1

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

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 1

[4] Tuomas Oikarinen, Subhro Das, Lam Nguyen, and Lily Weng. Label-free concept bottleneck models. In *ICLR*, 2023. 1

[5] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004. 1

[6] Chenming Shang, Shiji Zhou, Hengyuan Zhang, Xinzhe Ni, Yujiu Yang, and Yuwang Wang. Incremental residual concept bottleneck models. In *CVPR*, 2024. 1

[7] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. 1

[8] Yufei Xu, Jing Zhang, Qiming Zhang, and Dacheng Tao. Vitpose: Simple vision transformer baselines for human pose estimation. In *NeurIPS*, 2022. 1

[9] Weiyu Zhang, Menglong Zhu, and Konstantinos G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013. 1