Supplementary Materials
HACS: Human Action Clips and Segments Dataset for Recognition and Temporal Localization

1. Index

The supplementary materials are organized as follows.

- **Video De-duplication.** We elaborate on the de-duplication process in building the HACS dataset in Sec. 2.
- ***HACS Clips* Annotation.** We present more details on clip annotation process and results in Sec. 3, including a demo of clips annotation tool. It can be accessed at *clips_annotation_tool.mp4*.
- ***HACS Segments* Annotation.** We present more details on segment annotation process and results in Sec. 4, including a live demo. It can be accessed at *segments_annotation_tool.mp4*.
- **Annotation Guideline**. A detailed annotation guideline is discussed in Sec. 5.
- **Action Classification on HACS Clips.** We show action classification results on *HACS Clips* in Sec. 6.
- **Transfer Learning on Action Localization.** We present transfer learning results on action localization in Sec. 7.
- **Exploiting Negative Examples in HACS Clips for Action Proposal Generation.** We validate the effectiveness of exploiting 1M negative examples in *HACS Clips* to improve action proposal generation in Sec 8.

2. Video De-duplication

Internal De-duplication. We de-duplicate videos within HACS, since YouTube may include several copies of the same video, possibly differing in post-processing steps, such as saturation/contrast enhancement. We use a method similar to that in [1], and compute a similarity score between HACS videos. A video is removed if the score is above the threshold. In this step, we remove $171K$ duplicate videos.

External De-duplication. We perform extra de-duplication to ensure that HACS does not overlap with the validation and test sets of other video datasets, including Kinetics, ActivityNet, UCF101 and HMDB51. Similarity scores between HACS videos and other datasets' videos are computed, and the HACS videos are removed when scores are above the threshold. We sequentially de-duplicate HACS with respect to Kinetics, ActivityNet, UCF101 and HMDB51, and remove 4029, 1925, 660 and 11 videos from HACS.

3. HACS Clips Annotation

3.1. Person Detection-based Preprocessing

In this preprocessing step, we use a person detector to remove video clips that do not contain people. To reduce the computation, a simple shot detection based on color histogram distance between consecutive video frames is performed beforehand to segment the video into shots. For each shot, 2 frames are sampled and a person score is computed. We run the Faster R-CNN [3] trained on the COCO person class to get the maximum score of person being present over the two frames. A threshold of 0.5 on the person score is used to find shots with people (nearly 49.7% of all video shots). These shots form the initial proposals of human action clips for further processing. To validate the hypothesis that video clips without person present can be considered as negative sample and effectiveness of our person detector, we evaluate the same person detector on the validation set of ActivityNet-v1.3 where the ground truth of action segments are available. The recall of action clips is 96.9%, which indicates the majority of action clips indeed have people present . If all shots in a video have scores below the threshold, we remove such video. We remove $193K$ videos in this step.

3.2. Clip Annotation Tool Demo

clips_annotation_tool.mp4 is a live demo of clip annotation process, and it demonstrates annotating clips using our tool is highly efficient. As shown in the demo video, a number of clips are presented simultaneously, and the annotator

only needs to click the clips to flip their labels, which are indicated by boxes in green (positive) and red (negative), respectively.

4. HACS Segments Annotation

We have included a live demo of using our action segment annotation tool to label the action segment boundaries in the video (*segments_annotation_tool.mp4*). As shown in Figure 5 of the paper, a timeline overview is shown below the video player, and a zoom-in view of current time window is shown in the bottom for accurate temporal annotation. Detailed annotation instructions are given below.

- **Create a segment**. To start a segment, first move the video to the exact point where you want a segment started. Only then press 'z' to start the segment. You'll notice that the dashed line became solid. Now, scroll the video to the exact point where you want to end the segment. Only then press 'z' to end the. segment. You'll notice that the solid line became dashed again.
- **Modify an existing segment**. There is currently no way to modify an existing segment. Try to scroll the video to the exact point where the segment starts/ends before pressing 'z'. This should be enough to mark segments accurately. If you made a mistake, you can always delete a segment and draw it again.
- **Delete a segment**. Click on the segment in the Annotation chart. Delete button will appear under the segment.
- **Submit the annotation and move to next video**. Click Submit button.
- **Reject to annotate a video**. If the video is problematic, (e.g. too blurring, too dark, not playable), you can click 'Reject' to skip it and move to the next video.
- **Resume to work on other videos of a specific action in the queue**. Access the queue for that specific action and continue the annotation.

5. Annotation Guideline

We prepare an annotation guideline for both clip and segment annotations. It clarifies ambiguity in the annotation process. For each action, we give (1) textual action definition for clip and segment annotations, (2) positive clip examples, and (3) optionally hard negative clip examples to clarify potential ambiguity. Before annotating clips, annotators are required to receive an 1-hour training session where they carefully go over the annotation guideline and learn how to use the annotation tool. The full annotation guideline can be found in the homepage of HACS dataset.

Pretraining	AR@100	AUC
None	63.52	53.02
HACS Segments	69.31	61.93

Table 1: Transfer learning on action proposal generation. Results of TAG model pretrained on HACS Segments and tested on the ActivityNet validation set.

Note that in the original guideline, positive and negative examples are shown as videos, we replace them with static images due to constraints of PDF document format.

6. Action Classification on HACS Clips

In Table 2 of the paper, we reported the classification accuracy of I3D models on the validation set of *HACS Clips*. In this section, we show the class-wise accuracy of I3D model using RGB input, as well as the distribution of positive and negative clips per action class in Figure 1, where actions are sorted by the number of positive clips.

Action *Playing violin* has the most positive clips (10.3K) while *Preparing Pasta* has the fewest positive clips (149). On one side, actions with sufficiently many positive clips are more likely to attain high accuracy. The average accuracy of top 50 actions is 93.2%, while it is only 66.3% for the last 50 actions. On the other side, a large number of positive clips do not always warrant high accuracy. Actions *Making a cake* and *Rollerblading* rank 9th and 20th by the number of positive clips. However, they only achieve the 143th and 135th best accuracy. We hypothesize those actions are not yet well modeled by the current method, and need further investigation. For instance, action *Making a cake* has large intra-class variations due to the various steps in making a cake. In contrast, action *Removing ice from car* rank 197th, but achieves the 48th best accuracy, since the action has simple motion pattern of wiping the car and strong contextual cue (i.e. ice on the car).

7. Transfer Learning on Action Localization

In this section, we show how pre-training on HACS Clips and HACS Segments can help two related tasks: action proposal generation, and action localization.

7.1. Action Proposal Generation

Table 1 shows a comparison of results of TAG method on ActivityNet-1.3 for action proposal generation. By pretraining on HACS Segments, we can greatly improve the quality of action proposals, by 5.79% on AR@100, and 8.91% on AUC.

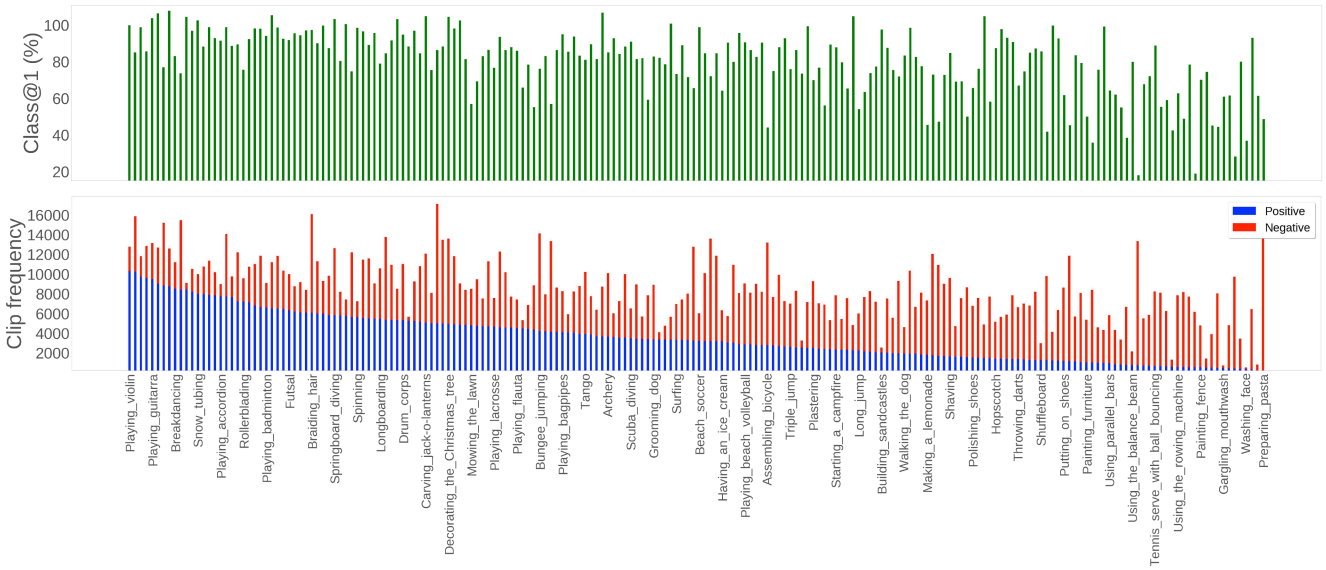

Figure 1: The performance of I3D model on the validation set of *HACS Clips*. **Top**: Class-wise accuracy. **Bottom**: The distribution of positive and negative clips over actions classes.

mAP @ IoU	0.3	0.4	0.5	0.6	0.7	Average
learning from scratch	33.0	27.1	18.9	10.5	4.0	18.7
pre-train on Sports1M [4]	40.1	29.4	23.3	13.1	7.9	22.8
pre-train on *HACS Clips*	**45.0**	**35.8**	**29.2**	**17.2**	**9.5**	**27.3**

Table 2: Transfer learning on action localization. Results of CDC model pretrained on HACS Clips and test on the THUMOS14 test set.

mAP @ IoU	0.5	0.60	0.70	0.80	0.90	0.95	Average
w/o refinement [8]	43.3	36.7	8.7	1.9	0.2	0.0	15.3
w/ refinement, no pre-training	43.0	**36.0**	**28.6**	13.5	1.4	0.2	22.4
w/ refinement, pre-train on Sports1M[4]	**45.3**	N/A	N/A	N/A	N/A	0.2	23.8
w/ refinement, pre-train on *HACS Clips*	42.9	34.7	28.0	**20.9**	**9.7**	**1.4**	**24.9**

Table 3: Transfer learning on action localization. Results of CDC model pretrained on HACS Clips and test on the ActivityNet validation set.

7.2. Action Localization

Further we present results of transfer learning for action localization task with *HACS Clips* and *HACS Segments*, respectively.

7.2.1 Pre-training on HACS Clips

Framework. We adopt the Convolutional-De-Convolutional (Conv-Deconv) framework in [4]. Briefly speaking, candidate video segments are proposed from the untrimmed videos. After that, dense frame-level labels are predicted for all the proposal segments to localize actions accurately. Note that we only use RGB frames (without optical flow) as input in all experiments of action localization. Please refer to [4] for more details on the framework.

Network Architecture. To produce dense frame-wise prediction, Conv-Deconv operators are used. Given input RGB clip of size $32 \times 3 \times 112 \times 144$, the localization model uses a Res3D-34 trunk, which produces feature maps of size $4 \times 512 \times 7 \times 7$, and 3 successive Conv-Deconv operators, which produce feature maps of size $32 \times 512 \times 1 \times 1$. A 201-way softmax is used to generate predictions over 200 actions and 1 background class.

Pretraining	0.50	0.75	0.95	Average
None [60]	39.12	23.48	5.49	23.98
HACS Segments	41.65	26.49	6.58	25.81

Table 4: Transfer learning on action localization. Results of SSN [9] pretrained on HACS Segments and test on the ActivityNet-1.3 validation set.

Training. For pre-training, we randomly sample 32 frames from 2-second *HACS clips*, and give all frames a uniform label based on clip annotation. The pre-trained model is further finetuned on THUMOS14 and ActivityNet. For each action segment annotation, we expand it by 2 seconds at both the beginning and the end of the segment. Then we randomly choose 32 frames as training examples. We assign positive action label to frames within the action segment, and the background label to frames outside of action segments.

Inference. The localization model can produce frame-level dense predictions. However, it is non-trivial to design a robust algorithm that leverages them to generate the temporal extent of action segments. On the other hand, existing segment proposal methods are able to generate candidate segments with high recall. Therefore, for THUMOS14, we adopt proposals from [5], and apply our dense prediction model for localization. For ActivityNet, we adopt the output from [8] as proposals, and then we follow the method in [4] to refine proposal boundaries. Please refer to [4] for the details of boundary refinement algorithm.

Results. Results on THUMOS14 are provided in Table 2 and those on ActivityNet are given in Table 3. In general, pre-training significantly improves mean average precision (mAP) of localization models. Specifically, models pre-trained on *HACS Clips* are good at improving hits at high IoU thresholds (IoU>0.6), which can be explained by the more accurate action boundary prediction. Compared with the localization model based on Res3D-34 and Conv-Deconv operators trained from scratch, the same model pre-trained on *HACS Clips* achieves an absolute gain of 8.6% on THUMOS14 and of 2.5% on ActivityNet. We also compare with the original Conv-Deconv work pre-trained on Sports-1M [6]. Our model outperforms it by 4.5% (22.8% Vs 27.3% on THUMOS14 and 1.1% (23.8% Vs 24.9%) on ActivityNet.

7.2.2 Pre-training on HACS Segments

We can also use HACS Segments to pre-train action localization models, such as SSN [9], and finetune them on ActivityNet-v1.3 benchmark. Results of SSN [9] are reported in Table 4. Compared to the SSN learned from scratch, the model pretrained on HACS Segments yields an average mAP boost of 1.83%.

TSN [7] Training Set	AR@10	AR@100	AUC
Positive examples only	36.96	62.17	52.19
All examples	**39.13**	**63.62**	**53.41**

Table 5: Comparing results of BSN [2] method on *HACS Segments* dataset. We train TSN models on annotated clips in HACS Clip to extract snippet-level features.

8. Exploiting Negative Examples in HACS Clips for Action Proposal Generation

In Section 5.1 of the paper, we discussed on exploiting negative action clips for improving action proposal generation. More details are presented below.

In *HACS Clips*, there are 1M negative clips. The proposed clip sampling method leads to many hard negative examples, such as clips where both person and context are present, but action is not being performed.

To understand how they can help learn more useful features for action proposal generation, we train two TSN models using only positive examples and all examples in *HACS Clips*, respectively. The TSN model trained only on positive examples outputs 200-D probability vector. After concatenating features from RGB model and flow model, snippet-level feature is 400-D. Results are reported in Table 5. BSN using snippet-level features from TSN models trained on both positive and negative examples moderately improves both AR@100 and AUC score. It verifies the importance of explicitly modeling the background class, which helps BSN to better predict the scores of a snippet being the start, course and end of action segments.

References

[1] 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

[2] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *European conference on computer vision*, 2018. 4

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1

[4] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: Convolutional-deconvolutional networks for precise temporal action localization in untrimmed videos. In *The IEEE Conference on Com-*

puter Vision and Pattern Recognition (CVPR), July 2017. 3, 4

[5] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1049–1058, 2016. 4

[6] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 4

[7] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 4

[8] Ruixin Wang and Dacheng Tao. UTS at ActivityNet 2016. In *Computer Vision and Pattern Recognition Workshop (CVPRW) on ActivityNet Large Scale Activity Recognition Challenge*, 2016. 3, 4

[9] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017. 4