Learning To Recognize Procedural Activities with Distant Supervision

Supplementary Material

Xudong Lin[1*] Fabio Petroni[2] Gedas Bertasius[3]
Marcus Rohrbach[2] Shih-Fu Chang[1] Lorenzo Torresani[2,4]
[1]Columbia University [2]Facebook AI Research [3]UNC Chapel Hill [4]Dartmouth

A. Implementation Details

Our implementation uses the wikiHow articles collected and processed by Koupaee and Wang [8], where each article has been parsed into a title and a list of step descriptions. We use a total of $S = 10,588$ steps collected from the $T = 1059$ tasks used in the evaluation of Bertasius *et al*. [3]. This represents the subset of wikiHow tasks that have at least 100 video samples in the HowTo100M dataset. We note that the HowTo100M videos were collected from YouTube [1] by using the wikiHow titles as keywords for the searches. Thus, each task of HowTo100M is represented in the knowledge base of wikiHow, except for tasks deleted or revised.

We implement our video model using the code base of TimeSformer [3]. All methods and baselines based on TimeSformer start from a configuration of ViT initialized with ImageNet-21K ViT pretraining [5]. Each segment consists of 8 frames uniformly sampled from a time-span of 8 seconds. For pretraining, we sample segments according to the ASR temporal boundaries available in HowTo100M. If the time-span exceeds 8 seconds, we sample a segment randomly within it, otherwise we take the 8-second segment centered at the middle point. For our pretraining of TimeSformer on the whole set of HowTo100M videos, we use a configuration slightly different from that adopted in [3]. We use a batch size of 256 segments, distributed over 128 GPUs to accelerate the training process. The models are first trained with the same optimization hyper-parameter settings for 15 epochs as [3]. Then the models are trained with AdamW [10] for another 15 epochs, with an initial learning rate of 0.00005. The pretraining of our model took 55 hours using 128 GPUs. As a reference, Miech et al. [11] report that pretraining S3D with MIL-NCE on HowTo100M required 3 days with 64 8-core TPUs.

To perform classification of multi-step activities as well as step forecasting on downstream datasets we use a single basic transformer layer [14] trained on top of our fixed embeddings. The transformer layer has 768 embedding dimensions and 12 heads. The step embeddings extracted with TimeSformer are augmented with learnable positional embeddings before being fed to the transformer layer. We train the transformer layer on sequences of 8 embedding vectors extracted from a series of 8 adjacent 8-second clips from the input video (spanning a total of 64 seconds).

For step classification, we train a simple linear classifier on embeddings extracted from individual segments of the downstream dataset. If the segment exceeds 8 seconds we sample the middle clip of 8 seconds, otherwise we use the given segment and sample 8 frames from it uniformly.

For egocentric video classification on EPIC-KITCHENS-100, we follow the experimental setup described in [2], except that we sample 32 frames as input with a frame rate of 2 fps to cover a longer temporal span of 16 seconds.

For the downstream tasks of procedural activity recognition, step classification, and step anticipation, we train the extra layer(s) on top of the frozen step embedding representation for 75K iterations, starting with a learning rate of 0.005. The learning rate is scaled by 0.1 after 55K and 70K iterations, respectively. The optimizer is SGD. We ensemble predictions from 4, 3, and 4 temporal clips sampled from the input video for the three tasks, respectively. We follow [9, 13] to split the data sets into a training set and a test set on the COIN and the Breakfast dataset, respectively.

B. Classification Results with Different Number of Transformer Layers

In the main paper, we presented results for recognition of procedural activities using as classification model a single-layer Transformer trained on top of the video representation learned with our distant supervision framework. In Table 1 we study the potential benefits of additional Transformer layers. We can see that additional Transformer layers in the classifier do not yield significant gains in accuracy. This suggests that our representation enables accurate classifica-

*Research done while XL was an intern at Facebook AI Research.

# Transformer Layers	Acc (%) of Basic Transformer	Acc (%) of Transformer w/ KB Transfer
0 (Avg Pool)	81.0	n/a
0 (Concat)	81.5	n/a
1	88.9	90.0
2	90.0	89.8
3	89.3	90.4

Table 1. Effect of different number of Transformer layers in the classification model used to recognize procedural activities in the COIN dataset. The classifier is trained on top of the video representation learned with our distant supervision framework.

tion of complex activities with a simple model and does not require additional nonlinear layers to achieve strong recognition performance. We also show the results without any transformer layers, by training a linear classifier on the average pooled or concatenated features from the pretrained TimeSformer. It has a substantially low results compared to using transformer layers for temporal modeling, which indicates that our step-level representation enables effective powerful temporal reasoning even with a simple model.

C. Representation Learning with Different Video Backbones

Although the experiments in our paper were presented for the case of TimeSformer as the video backbone, our distant supervision framework is general and can be applied to any video architecture. To demonstrate the generality of our framework, in this supplementary material we report results obtained with another recently proposed video model, ST-SWIN [15], using ImageNet-1K pretraining as initialization. We first train the model on HowTo100M using our distant supervision strategy and then evaluate the learned (frozen) representation on the tasks of step classification and procedural activity classification in the COIN dataset. Table 2 and Table 3 show the results for these two tasks. We also include results achieved with a video representation trained with full supervision on Kinetics as well as with video embeddings learned by k-means on ASR text. As we have already shown for the case of TimeSformer in the main paper, even for the case of the ST-SWIN video backbone, our distant supervision provides the best accuracy on both benchmarks, outperforming the Kinetics and the k-means baseline by substantial margins. This confirms that our distant supervision framework can work effectively with different video architectures.

D. Action Segmentation Results on COIN

In the main paper, we use step classification on COIN as one of the downstream tasks to directly measure the quality of the learned step-level representations. We note that some prior works [11, 17] used the step annotations in COIN to evaluate pretrained models for action segmentation. This task entails densely predicting action labels at each frame. Frame-level accuracy is used as the evaluation metric. We argue that step classification is a more relevant task for our purpose since we are interested in understanding the representational power of our features as step descriptors. Nevertheless, in order to compare to prior works, here we present results of using our step embeddings for action segmentation on COIN. Following previous work [11,17], we sample adjacent non-overlapping 1-second segments from the long video as input to our model. We use our model pretrained on HowTo100M as a fixed feature extractor to obtain a representation for each of these segments. Then a linear classifier is trained to classify each segment into one of the 779 classes (778 steps plus the background class). Our method achieves a frame accuracy of 67.6%. The representation learned with full-supervision using action labels in Kinetics gives a substantially lower accuracy: 63.8% with the same classification model as our method. The methods in [11,17] achieve an accuracy of 57.0% and 61.0%, respectively. Using the same linear setup as our model, VideoCLIP features [16] pretrained on HowTo100M achieve an accuracy of 59.9%, i.e., 7.7% lower than our representation.

E. More Qualitative Results and Discussion

Visualization of Distant Supervision. In Figure 1 we provide visualizations of steps assigned by our distant supervision method for three video examples. We can observe that the matched step descriptions capture high-level semantics about actions and objects, which are conversely often missed by the narrations. An example is given in Figure 1a where the narration in the last segment ("really really hot water you can use") does not correspond to an object or an action directly recognizable in the segment. The language model assigns this narration to a more expressive step description (*Open the hot water faucet in your sink or tub*). Figure 1b shows that the assigned steps capture higher level information compared to traditional atomic actions. For example, a video segment of pouring oil into a heated pan is matched to *Prepare to fry tortillas*.

Visualization of Step Classes. In order to better understand the variety of video segments that are grouped under a given step, in Figure 2 we show three video clips assigned to three given steps. We can observe that our method can successfully group together video segments that are coherent in terms of the demonstrated step. Note that, at the same time, the segments assigned to a given step exhibit large variations in terms of appearance (e.g., color, viewpoint, object instances). Because our model assigns segments to step descriptions purely based on language information, it is insensitive to these large appearance variations. This invariance is then transferred to the video model: by using these distantly supervised step classes as supervision for the

Segment Model	Pretraining Supervision	Pretraining Dataset	Acc (%)
ST-SWIN	Supervised: action labels	Kinetics	44.0
ST-SWIN	Unsupervised: k-means on ASR	HT100M	44.8
ST-SWIN	**Unsupervised: distant supervision (ours)**	HT100M	**50.3**

Table 2. Comparison to baselines for the problem of step classification on the COIN dataset using ST-SWIN as video architecture.

Long-term Model	Segment Model	Pretraining Supervision	Pretraining Dataset	Acc (%)
Basic Transformer	ST-SWIN	Supervised: action labels	Kinetics	79.6
Basic Transformer	ST-SWIN	Unsupervised: k-means on ASR	HT100M	82.4
Basic Transformer	**ST-SWIN**	**Unsupervised: distant supervision (ours)**	HT100M	**88.3**

Table 3. Comparison to baselines for the problem of classifying procedural activities on the COIN dataset using ST-SWIN as video architecture.

video model, our method trains the video representation to be invariant to these appearance variations and to capture the higher-level semantics represented in each step class.

Limitations and Failure Cases. Our approach may fail in assigning the correct step to a segment due to errors caused by the language model and due to excessive noise or ambiguity in the ASR sentence. Furthermore, the step description may refer to actions or objects not represented in the segment. For example, in Figure 2c the assigned step *Put in the oven and bake until the cheese is melted* provides an accurate semantic description for the segments, but it refers to an object ("oven") that is not shown in the video frames. On one hand, this visual misalignment may render the training difficult; on the other hand, it may still be beneficial, since it forces the model to use contextual information (e.g., visible objects that tend to co-occur with "oven", such as the bakeware objects appearing in the frames) to recognize the high-level semantics of the steps. Another potential limitation is the temporal misalignment between speech and visual content. However, this problem can be reduced by expanding the temporal span of the ASR text to increase the probability of including the relevant text for the given video segment, or by adopting a multiple-instance learning scheme [11] to find the correct temporal alignment between ASR text sentences and video segments.

Complexity of Steps. In our experiments we demonstrated that, on the downstream problems of step and task classification, our distantly-supervised video representation outperforms video descriptors trained with full supervision on traditional action classes. We hypothesize that this is due to the fact that each step typically consists of multiple actions performed in sequence, unlike traditional action classes which typically encode a single atomic action (e.g., "drinking", "jumping", "punching"). To assess this hypothesis we analyzed the number of verbs returned by the POS tagger [4] for each wikiHow step description as a measure of the complexity of the step. Figure 3 shows the distribution of the number of verbs. The average and the median number of verbs in step descriptions are 10.1 and 8.0, respectively. Furthermore, more than 85% of the steps contain at least 2 verbs. This indeed suggests that steps tend to have a higher-level of complexity compared to traditional atomic actions.

F. Further Details about Step Forecasting

We follow the training/validation split in the COIN dataset to train and evaluate our models for step forecasting. By constraining the observed history to contain at least one step, we construct a training set of 22037 samples and a validation set of 6721 samples. Fig. 4 shows the distribution of the gaps (in seconds) separating the history from the step to predict. The average and the median of the gap are 21 seconds and 14 seconds, respectively. Thus, the forecasting gaps in this benchmarks are substantially longer than those used in other action anticipation tasks [6, 7, 12]. This makes this benchmark particularly challenging as the model is asked to predict the step of segments far away in the future compared to the observed history.

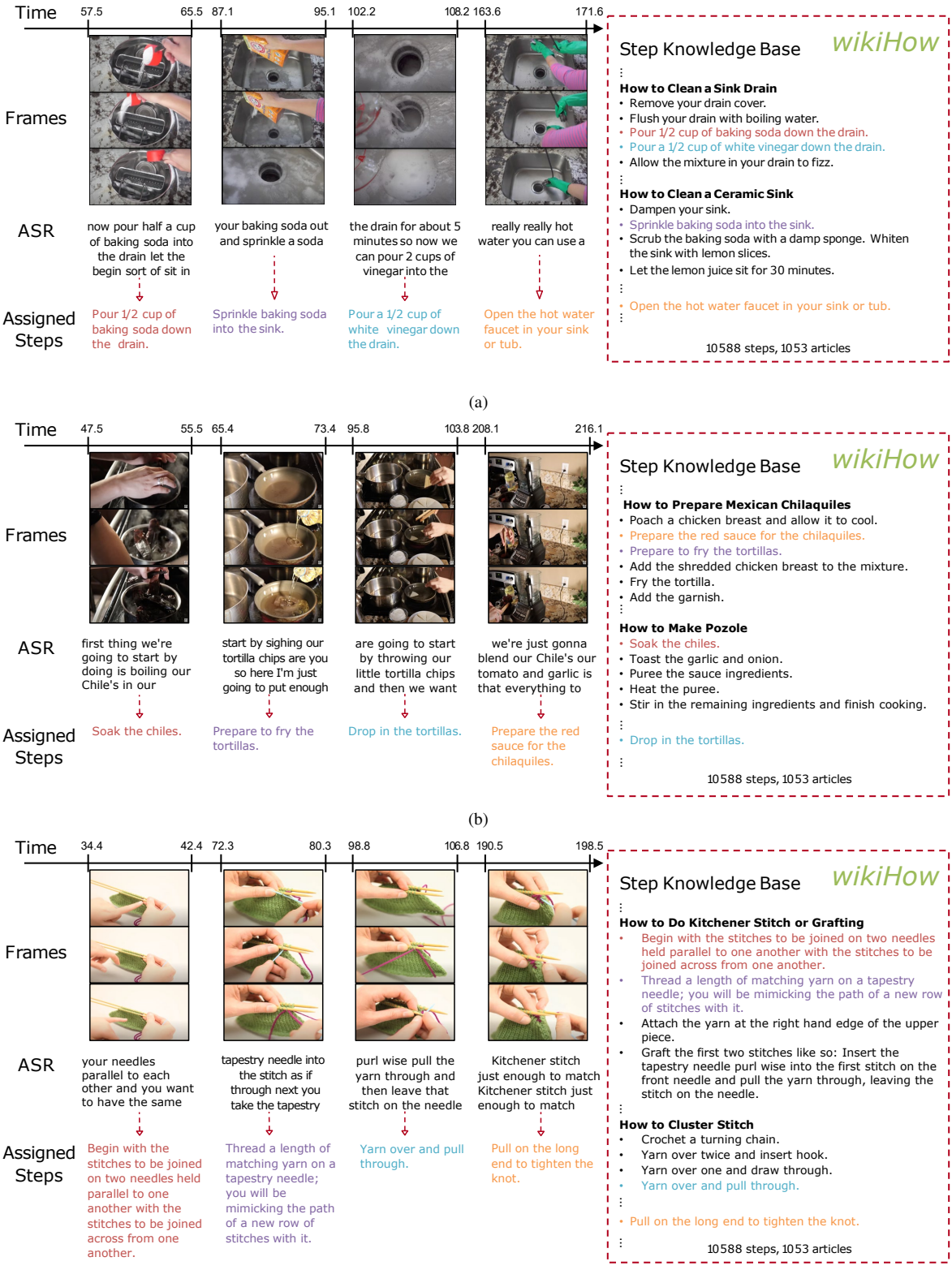

Figure 1. Visualization of steps assigned by our distant supervision method for three different video examples from the HowTo100M dataset. The assigned steps provide more expressive descriptions compared to the noisy ASR sentences.

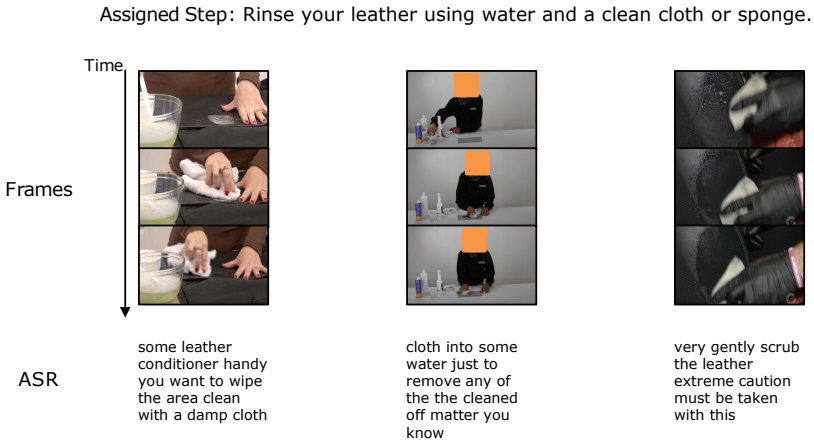

Figure 2. Visualization of segments assigned to three given steps by our distant supervision framework. We can observe that our method successfully groups together segments that are semantically coherent under each step, despite their large differences in appearance. Faces in the frames are artificially masked for privacy reasons.

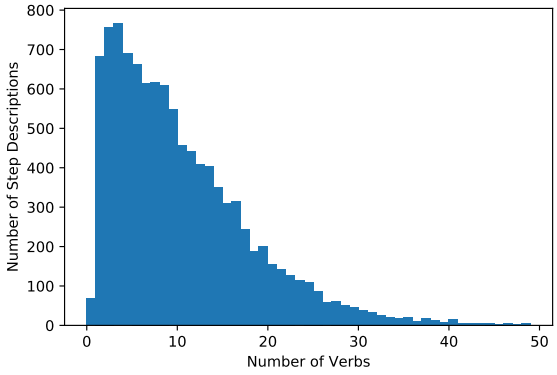

Figure 3. Histogram of number of verbs in wikiHow step descriptions.

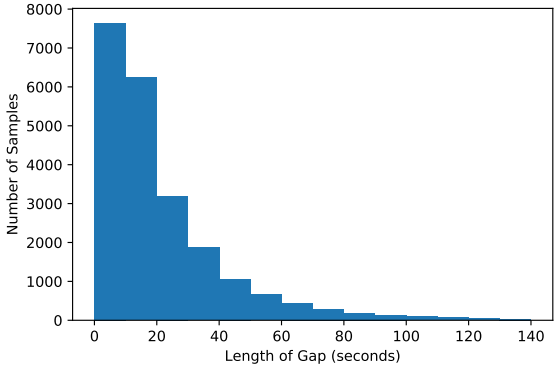

Figure 4. Histogram of the temporal gaps separating the observed history from the step to forecast in the COIN dataset.

References

[1] YouTube. https://www.youtube.com/. 1

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 1

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 1

[4] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009. 3

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[6] Harshala Gammulle, Simon Denman, Sridha Sridharan, and Clinton Fookes. Predicting the future: A jointly learnt model for action anticipation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3

[7] Minh Hoai and Fernando De la Torre. Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202, 2014. 3

[8] Mahnaz Koupaee and William Yang Wang. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*, 2018. 1

[9] Hilde Kuehne, Ali Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 780–787, 2014. 1

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 1

[11] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9879–9889, 2020. 1, 2, 3

[12] Michael S Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*, 2011. 3

[13] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1207–1216, 2019. 1

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1

[15] Jue Wang, Gedas Bertasius, Du Tran, and Lorenzo Torresani. Long-short temporal contrastive learning of video transformers. *arXiv preprint arXiv:2106.09212*, 2021. 2

[16] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. *arXiv preprint arXiv:2109.14084*, 2021. 2

[17] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2