# Supplementary Materials for Learning Procedure-aware Video Representation from Instructional Videos and Their Narrations

Yiwu Zhong[1]*, Licheng Yu[2], Yang Bai[2], Shangwen Li[2], Xueting Yan[2]†, Yin Li[1]†
[1]University of Wisconsin-Madison, [2]Meta AI
yzhong52@wisc.edu, {lichengyu, yangbai, dylanwen, xyan18}@meta.com, yin.li@wisc.edu

In this supplement, we describe (1) the derivation of our training loss, (2) the implementation details of data pre-processing, our model architecture, and key frame generation, (3) experiment results on the additional benchmark on COIN dataset, and (4) additional ablation studies on open-vocabulary recognition, the effects of using ASR phrases and backbone architecture. For sections, figures, tables, and equations, we use numbers (*e.g.*, Sec. 1) to refer to the main paper and capital letters (*e.g.*, Sec. A) to refer to this supplement.

## A. Derivation of Training Loss

Our method aims at minimizing the negative log likelihood $-\log p(Y|X)$ (Eq. 1 in paper). Here, we provide the derivation of its evidence lower bound, as shown in Eq. A, where $x_i$ are video embeddings learned by our video encoder $f(\cdot)$, $y_i$ are text embeddings offered by a pre-trained text encoder $g(\cdot)$ from CLIP [12] that remains fixed during our training. $\{x_i\}$ and $\{y_i\}$ are observed video and text embeddings, while $x_j$ and $y_j$ are the missing (masked) video and text embeddings.

There are three terms in the evidence lower bound, with each one corresponding to a loss in our main paper. First, $p(y_i|x_i)$ is computed by Eq. 6 of the paper, as a softmax over the cosine similarity between an input video embedding and a set of text embeddings. This term corresponds to the loss $\mathrm{L}_{\mathrm{X}E}$ (Eq. 8). Second, $p(x_j|\{x_i\}_{i\neq j})$ is approximated using a diffusion model that consists of a diffusion process and an reverse diffusion (denoising) process. This term is performed by the loss $\mathrm{L}_{MSE}$ (Eq. 9). Third, $p(y_j|x_j)$ seeks to predict text embedding $y_j$ using the masked video embedding $x_j$. It is again calculated by Eq. 6 of the paper. This term corresponds to $\mathrm{L}_{\mathrm{MC}}$ (Eq. 10).

## B. Additional Implementation Details

**Data Pre-processing**: During pre-training, we used the timestamps of ASR sentences to segment video clips from full videos. For step classification, the video clips are trimmed by human-annotated step boundaries. When evaluating step classification, multi-view augmentation is applied with 3 clips sampled on the temporal dimension. For step forecasting (both training and evaluation), we cropped 68 seconds of video before the target action and uniformly cut it into 8 video clips as the model input. For HowTo100M [10] and COIN dataset [13, 14], we sampled 1 frame per second. For EPIC-Kitchens-100 dataset [2], we sampled 16 frames per second. The text embedding of each verb phrase was the averaged embedding over 28 action prompts[1].

**Model Architecture and Hyper-parameters**: We adopted TimeSformer architecture [1] for our video encoder. TimeSformer is a Transformer [15] based model that applies attention mechanism over both spatial and temporal dimension. For denoising model, we used Transformer from CLIP's implementation[2] with bi-directional attention. In denoising model, we implemented the maximum time level $T$ as 4, maximum length of video sequence as 9, and the number of Transformer layers as 4. For time variable in diffusion model, we first mapped it into vector representation using position embeddings and then added it to the input of Transformer. When calculating the matching score between video and text embedding (Eq. 4 in main paper), we divided the matching score by a temperature $\tau = 0.02$ when computing the softmax.

**Details about Future Key Frame Generation**: Future key frame generation is posed as text guided image-to-image translation, where the text is provided by our predicted step and the image is from a sampled frame within the current video. Specifically, we use a pre-trained stable diffusion

*Work done while Yiwu Zhong was an intern at Meta.
†Co-corresponding authors.

[1]https://github.com/openai/CLIP/blob/main/data/prompts.md#kinetics700
[2]https://github.com/openai/CLIP

$$
\begin{aligned}
-\log p(Y|X) &= -\log(p(y_j|x_j)\cdot p(x_j|\{x_i\}_{i\neq j})\cdot \prod_i p(y_i|x_i)) \\
&\leq \underbrace{\sum_i -\log(p(y_i|x_i))}_{\text{cross-entropy loss } (\mathrm{L}_{XE})} \\
&+ \underbrace{\sum_{t=1}^{T} \mathbb{E}_{x_j^t\sim p(x_j^t|x_j^0,\{x_i\}_{i\neq j})}\left[\mathbb{D}_{KL}(p(x_j^{t-1}|x_j^t,x_j^0,\{x_i\}_{i\neq j})||p_\theta(x_j^{t-1}|x_j^t,\{x_i\}_{i\neq j}))\right]}_{\text{diffusion model loss } (\mathrm{L}_{MSE})} \\
&+ \underbrace{\mathbb{E}_{x_j\sim p_\theta(x_j|\{x_i\}_{i\neq j})}\left[-\log(p(y_j|x_j))\right]}_{\text{cross-entropy loss } (\mathrm{L}_{MC})}
\end{aligned}
\tag{A}
$$

| | Model | Pretraining | | Top-1 |
|---|---|---|---|---|
| | | Supervision | Dataset | Acc. (%) |
| 1 | TSN (RGB+Flow) [13] | Supervised: action labels | Kinetics | 73.4* |
| 2 | S3D [16] | Unsupervised: ASR w. MIL-NCE [9] | HT100M | 70.2* |
| 3 | SlowFast [3] | Supervised: action labels | Kinetics | 71.6 |
| 4 | TimeSformer [1] | Supervised: action labels | Kinetics | 83.5 |
| 5 | ClipBERT [5] | Supervised: captions | COCO+VG | 65.4 |
| 6 | VideoCLIP [17] | Unsupervised: ASR | HT100M | 72.5 |
| 7 | TimeSformer [1] | Unsupervised: ASR w. MIL-NCE [9] | HT100M | 85.3 |
| 8 | DistantSup [7] | Unsupervised: ASR + wikiHow | HT100M | 88.9 |
| 9 | Ours | Unsupervised: ASR | HT100M | **90.8** |

Table A. Procedural activity classification on COIN dataset. * indicates the model is fully fine-tuned on COIN dataset.

model[3] and employ SDEdit [8]. SDEdit adds noise to the sampled input video frame, and then denoises the resulting image using stable diffusion model and the text of our predicted step, in order to generate a future video frame.

# C. Additional Benchmarks

## C.1. Procedural Activity Classification

We follow the benchmark in DistantSup [7] to evaluate procedural activity recognition on COIN with top-1 accuracy reported. Given a video that has recorded multiple steps, the model classifies the entire video into an activity category (e.g., "make coffee"). Similar to step forecasting, we only fine-tune the diffusion model to predict activity category, with the frozen video encoder as a feature extractor.

In Table A, we compare our model with a series of baselines as in DistantSup [7], such as SlowFast [3], TimeSformer [1] and S3D [16]. These baselines are pre-trained by either human-annotated action labels or video ASR sentences. Our closest competitor is DistantSup [7] which learns individual action concepts by leveraging an external text knowledge base (wikiHow). Our model clearly outperforms all baseline models by a large margin (*e.g.*, **+1.9** over DistantSup in L8). Our experimental results suggest that our order pre-training approach, which captures the order among steps, can also improve the recognition of the entire sequence of steps, even if it was not designed for this task.

[3]https://github.com/CompVis/stable-diffusion

| COIN steps | Step descriptions during pre-training |
|---|---|
| fry eggs | fry chicken, lay eggs |
| calibrate the liquid | calibrate the meter |
| scrub the bathtub | clean the bathroom |
| chase for the frisbee | – |
| knead the meat | cut the meat, cook the meat |
| bake pizza | bake soda, bake powder, make pizza |
| put the sheet on the bed | take a sheet, make a sheet |
| melt the wax with water | melt the plastic, melt the cheese, put wax |
| wet and wash the hair | moisturize hair, rinse hair, wet my brush |
| place light into pumpkin | place your lights, adjust the light |

Table B. Visualization of step concepts. We show COIN steps (left) and the step descriptions in pre-training (HowTo100M) that have common verb/noun (right).

# D. Additional Ablation Studies

We present additional ablation studies on our model. The experiment settings follow the ablation study in the main paper, unless otherwise noticed.

**Can our model identify open-vocabulary step concepts?** Part of our learning objective is to match the video representations with text embeddings. Such a design allows

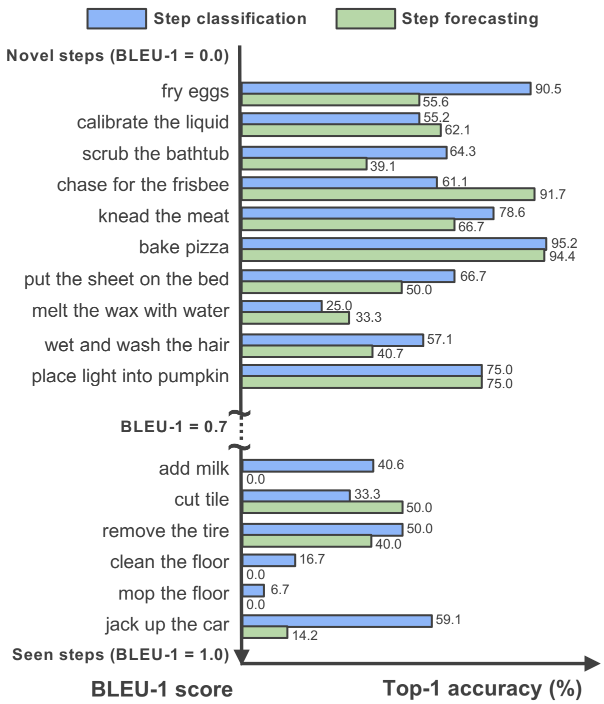


Figure A. Per-category top-1 accuracy for zero-shot step classification and forecasting. We rank the step concepts in COIN dataset by calculating its maximum BLEU-1 score [11] versus all step descriptions used in pre-training.

our model to support zero-shot recognition as we demonstrated in the paper. One natural question is how well our model performs during zero-shot recognition when facing step concepts that have not been seen during pre-training.

Figure A measures the overlap between step concepts during pre-training (from ASR results on HowTo100M) and during zero-shot recognition (from human-annotated categories on COIN), and reports per-category results for both seen and novel step categories. Specifically, we adopt BLEU-1 score [11] to match the step concepts, and report per-category top-1 accuracy for zero-shot step classification and forecasting. BLEU-1 score as zero indicates the novel steps and BLEU-1 score as one suggests that the exact steps have been seen during pre-training. In addition, we show the steps that have a common verb/noun as COIN steps in Table B.

We find that our model achieves high accuracy even if facing novel steps, *i.e.* the steps have low BLEU-1 score (*e.g.*, 90.5% for "fry eggs"). Further, we compute the top-1 accuracy for the steps with high BLEU-1 scores (*e.g.*, $\geq 0.7$) and the steps with low BLEU-1 scores (*e.g.*, $< 0.7$). These two groups include 103 and 675 steps, respectively, and have close top-1 accuracy across tasks (*e.g.*, 15.9 vs. 16.7 for step classification, 14.2 vs. 10.9 for step forecasting). These results suggest that our model is not limited

| Source | Zero-shot | | Fine-tuning | |
|---|---|---|---|---|
| | Classification | Forecasting | Classification | Forecasting |
| wikiHow sentences | 11.6 | 8.3 | 48.6 | 38.0 |
| ASR phrases | 11.8 | 9.0 | 47.8 | 38.9 |

Table C. Ablation study on different sources of step descriptions. Top-1 accuracy (%) on COIN dataset is reported. All models are pre-trained on a subset of HowTo100M dataset, defined by [1, 7].

| Source | Zero-shot | |
|---|---|---|
| | Classification | Forecasting |
| Ours (TimeSformer) | 16.6 | 11.3 |
| Ours (MViT-S) | 12.5 | 9.0 |

Table D. Ablation study on the different architectures of video encoder. All models are pre-trained on HowTo100M dataset.

to the step concepts considered in pre-training and supports open-vocabulary step recognition. We conjecture that our model has learned the components from similar phrases (*e.g.*, "fry chicken" and "lay eggs" shown in Table B), by learning to project video embeddings into the semantic space defined by the text embeddings of CLIP.

**Are ASR phrases sufficient to learn step concepts?** We propose to use the step phrases parsed from video ASR sentences for learning step concepts. The latest work DistantSup [7] found that external text corpus for procedure activities (*e.g.*, wikiHow [4]) can largely reduce the noise in ASR sentences. In this section, we explore using wikiHow sentences to pre-train our model.

In Table C, we compare our model with a variant pre-trained using wikiHow sentences, following [7]. Our results demonstrate that ASR phrases are sufficient to achieve competitive results across tasks and settings (*e.g.*, +0.7/+0.9 for step forecasting across zero-shot and fine-tuning settings). In other word, our model only requires ASR phrases generated from audio transcriptions of videos, without the need of an external text corpus describing the procedural activities as in [7].

**Backbone Architecture of Video Encoder**. In Table D, we study the effects of backbone architectures for our video encoder. We replace the default backbone TimeSformer with MViT-S [6] which is also a widely-used architecture for video encoders. We slightly increase the frame sampling rate of MViT-S from the default value of 4 to 6 so that the encoder can take a longer video (e.g., on COIN, the average duration of a step is 14 seconds). TimeSformer consistently outperforms MViT-S across tasks (*e.g.*, +4.1 on step classification). We conjecture that TimeSformer, which samples 8 frames from consecutive 256 frames, is better suited for recognizing actions with long durations, such as COIN steps. Conversely, MViT-S, which samples 16 frames from consecutive 96 frames, may perform better for recognizing actions with short durations and high-speed motion.

## References


[1] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021. 1, 2, 3

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The EPIC-KITCHENS dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(11):4125–4141, 2021. 1

[3] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[4] Mahnaz Koupaee and William Yang Wang. WikiHow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*, 2018. 3

[5] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7331–7341, June 2021. 2

[6] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. MViTv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4804–4814, June 2022. 3

[7] Xudong Lin, Fabio Petroni, Gedas Bertasius, Marcus Rohrbach, Shih-Fu Chang, and Lorenzo Torresani. Learning to recognize procedural activities with distant supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13853–13863, 2022. 2, 3

[8] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

[9] 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. 2

[10] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2630–2640, 2019. 1

[11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL)*, pages 311–318. Association for Computational Linguistics, 2002. 3

[12] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021. 1

[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, 2

[14] Yansong Tang, Jiwen Lu, and Jie Zhou. Comprehensive instructional video analysis: The COIN dataset and performance evaluation. *TPAMI*, 2020. 1

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[16] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018. 2

[17] 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