Weakly Supervised Video Representation Learning with Unaligned Text for Sequential Videos

Supplementary Material

A. Extra Experiment Studies

In this section, we present additional ablation studies about our method, including the effects of batch size, the number of clips sampled per video, the approach of extracting paragraph-level language representation, and sequence align loss.

A.1. Implementation Details

We implement our method with PyTorch. The vision backbone we employ is the pre-trained CLIP vision encoder based on ViT-B [1]. And the model is initialized adopting Kaiming and Xavier uniform initialization for different layers [4, 5]. In our module, the parameter of the vision backbone is unfrozen and finetuned when training. On the other hand, the language backbone is the pre-trained CLIP text encoder whose parameter is frozen totally. We split the raw video into 16 clips for a sequential video and randomly sample one raw frame from each clip in the training and uniformly sample frames in inference. The projection layer adopts a fully connected layer. The hidden layer dimension of transformer encoder [1] is 1024, and the depth is 2. The dimension of the video representations and paragraph representations is 512. The λ_1 in our model is equal to 1. The experiments are conducted on 4 NVIDIA 2080Ti GPUs with batch size 8. We adopt an AdamW optimizer [10] with cosine annealing learning rate scheduler with a base learning rate of 5×10^{-4}, and weight decay 0.01. More implementation details should be seen in the supplementary materials. And expect the experiment of sequence align loss to be conducted on the supervised sequence verification task, other experiments are conducted on the weakly supervised sequence verification task. We conduct all experiments on CSV dataset.

A.2. Batch size

To adapt to the change in batch size, we increase or decrease the learning rate exponentially. As Tab. 1 shown, our method achieves the best performance when the batch size is equal to 16. The larger the batch size, the more likely multiple videos of the same task will appear in the same mini-batch. Due to the limitation of GPU memory, the largest batch size can be set as 8 if we unfreeze the vision backbone.

A.3. Sampling

While changing the frames of sampling per video, the training time is doubled with the increasing number of frames. In this ablation study, all models have been training no more than 100 epochs or 12 hours on two GPUs due to the limitation of computing resources.

Method	Batch size	Frames	CSV
Ours	4	16	65.94
	8	16	67.46
	16	16	69.42
	24	16	69.21
	32	16	69.16

Table 1. Ablation studies of batch size on our proposed method

As Tab. 2 shown, when frames are set to 16, our model achieves the best performance. It is worth noting that, with more training steps (about twice the training time), the performance of 32 frames will increase to 68.20. However, we choose 16 as the default frame to balance batch size, number of frames, and training cost, we choose 16 as the default frame.

The significant reason for choosing sampling frames rather than video clips is the limitation of computational resources. Fine-tuning the full pre-trained backbone, such as VideoCLIP [9], is expensive. Similarly, to balance the efficiency of the network and fairly compare our method with CAT [7], we choose 16 frames as the same as CAT.

Method	Batch size	Frames	CSV
Ours	8	8	58.43
	8	16	67.46
	8	32	65.64
	8	48	64.40

Table 2. Ablation studies of the number of frames.

A.4. Paragraph feature

We design two ways to extract the feature of the paragraph. The one is concatenating all sentences into a paragraph description. Then we can obtain the paragraph-level representation by feeding the paragraph description into the language encoder. The other method is that feed individual procedure texts into the frozen language encoder to produce sentence representations and then obtain a paragraph-level representation by temporal mean pooling. The results shown in Tab. 3 illustrate that the method based on concatenation achieves better performance.

Method	Paragraph feature	CSV
Ours	pooling	67.07
	concat	67.46

Table 3. Ablation studies of the ways to extract paragraph features on our method.

A.5. Sequence alignment loss

For a fair comparison, some adjustments have been made to the architecture of our model on the supervised sequence verification task. Specifically, following [7], we apply the video sequence alignment mechanism to our model. Moreover, we also conduct experiments to investigate the effectiveness of using sequence alignment loss. We change the sequence align loss position to the last of the network. The results shown in Tab. 4 illustrate that sequence alignment loss L_{seq} could restrict the model to learning a better representation.

Method	L_{seq}	CSV
Ours	✗	84.47
	✓	84.69

Table 4. Ablation studies of the sequence alignment loss on our method.

B. Gumbel-Softmax with Viterbi

Due to the sum of the probabilities of each row cannot be greater than one and each probability value in a row should be the same, we simply set the value to $\frac{1}{N}$. As Eq. (1) shown, we set each element value in the upper diagonal matrix to $\frac{1}{N}$ and others to zero to keep the path of probability will be a one-way path.

$$A = \begin{bmatrix} \frac{1}{N} & \cdots & \frac{1}{N} \\ & \ddots & \vdots \\ 0 & & \frac{1}{N} \end{bmatrix}_{N \times N} \tag{1}$$

where A represents the Transition matrix of Viterbi algorithm [3].

C. TSM module

Following [2], we add the Temporal Similarity Matrix (TSM) module with residual connection to our vision module. In this ablation study, we only use the task classification loss L_{cls} instead of coarse-grained loss L_{coarse} and fine-grained loss L_{fine}. As Tab. 5 shown, we verify different similarity distances of TSM and residual connection types. And the experiments indicate that the TSM module with residual connection will improve the model performance.

However, as Tab. 6 shows, while we apply the TSM module to our method and train the model under weak supervision, the performance of the model degrades. It is reasonable that the model with the TSM module is not effective for language-video alignment tasks.

Method	Dist	Residual	CSV
CLIP [8]+TE [1]+MLP	✗	✗	77.35
	L2	add	77.42
	L2	concat	78.22
	Attn	add	76.89
	Attn	concat	77.71

Table 5. Ablation studies of the different kinds of TSM module on the baseline.

Method	L_{fine}	L_{coarse}	TSM	CSV
Ours	✓	✓	✗	79.80
	✓	✓	✓	76.00

Table 6. Ablation studies of the TSM on our proposed method.

D. Downstream tasks

D.1. Text-to-Video Matching

We validate the performance of the video-language representations on text-to-video matching, which aims to find the correct video corresponding to a sequence of texts from a series of videos. Specifically, we train our model on the CSV dataset under weak supervision and test it on our proposed benchmark about text-to-video matching. We calculate the similarity between each video representation V_i and paragraph representation L:

$$d = dis(L, V_i) \tag{2}$$

where $dis(.,.)$ represents the normalized Euclidean distance. And V_i represents $i_{th}(i \in [0, \ldots, 4])$ video representation. At last, we select the text-video pair with the max similarity.

CSV-Matching. To better evaluate the text-to-video matching, we rearrange the test set of CSV and propose a new scripted benchmark named CSV-Matching. It has 800 text-video pairs. Each text-video pair is composed of one sequence of text descriptions of procedures and five videos. All of the videos describe the same task but hold different procedures. There is only one correct video matching the text descriptions in each pair. CSV-test dataset contains 5 tasks and each task has 5 kinds of different procedures. We random select one kind of video from each procedure to compose one pairs. The benchmark and split script will be available.

Method	Backbone	Loss	Classification(Acc)
CAT [7]	ResNet-50 [2]	CLS, SEQ	61.08
CLIP [8]+TE+MLP	CLIP-ViT	CLS, SEQ	63.24
Ours(CLS)	CLIP-ViT	CLS, SEQ, Multi-grained loss	**69.57**

Table 7. Results of video classification on CSV.

Method	Backbone	Weakly supervised (w/o CLS)			Supervised (w CLS)		
		Def.	No Rep.	Rep.	Def.	No Rep.	Rep.
CAT [7]	ResNet50 [6]	47.70	57.82	49.99	51.13	63.25	45.96
CLIP [8]+TE+MLP	CLIP-ViT	50.83	65.28	53.73	48.50	65.21	51.25
Ours	CLIP-ViT	**52.55**	**68.98**	**56.16**	**59.57**	**77.78**	**54.95**

Table 8. Results of different methods on re-divided COIN-SV.

D.2. Video Classification

To demonstrate our method's transfer ability, we evaluate models in the downstream video classification task. We re-divided the CSV dataset for video classification task. The train set contains 689 videos and test set contains 185 videos. On the re-divided CSV test dataset (CSV-CLS), we evaluate representations of models with linear probing, which were pre-trained under weak supervision. As Tab. 7 shown, our method achieves better performance in the video classification task. The benchmark and split script will be available.

E. Limitations

While our method performs well on the major part of the data, there still are some failure cases. In realistic sequential videos, sub-actions are often repeated. In that case, there are multiple sentences with high similarity to a frame. It could mislead the model to generate biased pseudo-labels, which will lead to the deterioration of performance. For example, the occurrence of a large number of repetitive actions repetitive action might hidden achieving further performance.

The intuition of our fine-grained contrastive loss comes from a basic idea: *if the s_j is the corresponding sentence for frame h_i, the corresponding sentence for frame h_{i+1} is never before the s_j in sequence*. Due to a large number of repetitive actions, it might be difficult to achieve further performance. However, this method is still promising. As Tab. 8 shown, we have re-divided the COIN-SV test dataset based on whether existing repetitive actions in videos or not, which are COIN-SV-Rep (675 video pairs) and COIN-SV-NoRep (325 video pairs). In the original COIN-SV test dataset, there are 1000 video pairs for sequential video verification, built by 328 videos containing repetitive actions and 123 videos that do not. The results show that although the occurrence of repetitive actions will cause the deterioration of performance, our method can still achieve better results than other baselines. Moreover, the results conducted by our method may reflect the bias from the dataset.

References

[1] 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, 2

[2] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Counting out time: Class agnostic video repetition counting in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10387–10396, 2020. 2, 3

[3] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973. 2

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. 1

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[7] Yicheng Qian, Weixin Luo, Dongze Lian, Xu Tang, Peilin Zhao, and Shenghua Gao. Svip: Sequence verification for procedures in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19890–19902, 2022. 1, 2, 3

[8] 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 super-

vision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 2, 3

[9] 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. 1

[10] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019. 1