Expanding Language-Image Pretrained Models for General Video Recognition ———Supplementary Material———

Bolin Ni[1,2,*], Houwen Peng[4,†], Minghao Chen[5,*], Songyang Zhang[6], Gaofeng Meng[1,2,3,†], Jianlong Fu[4], Shiming Xiang[1,2], Haibin Ling[5]

[1] NLPR, Institute of Automation, Chinese Academy of Sciences
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] CAIR, HK Institute of Science and Innovation, Chinese Academy of Sciences
[4] Microsoft Research [5] Stony Brook University [6] University of Rochester

This supplementary material contains additional details of the main manuscript, and provides more experiment analysis. In Sec. 1, we present the details of our proposed architectures and the comparison methods. Next, we elaborate the hyperparameters in Sec. 2. Then, we overview the four datasets and provide the evaluation protocols of our experiments in Sec. 3. Finally, we provide more experiment analysis in Sec. 4.

1 Architecture Details

In this section, we elaborate the details of the proposed architectures in Sec. 1.1 and the comparison methods in the few-shot experiments in Sec. 1.2.

1.1 The proposed architectures

For CLIP, we provide three variants: X-CLIP-B/32, X-CLIP-B/16 and X-CLIP-L/14. In detail, there are three parts in our framework: a cross-frame communication transformer followed by a multi-frame integration transformer and a text encoder. X-CLIP-B/32 adopts ViT-B/32 (L_c=12, N_h=12, d=768, p=32) as parts of the cross-frame communication transformer, X-CLIP-B/16 uses ViT-B/16 (L_c=12, N_h=12, d=768, p=16), while X-CLIP-L/14 employs ViT-L/14 (L_c=24, N_h=12, d=1,024, p=14), where L_c denotes the layers, N_h refers to the number of attention heads, d represents the embedding dimension and p is the patch size. We use a simple 1-layer multi-frame integration transformer for all three X-CLIP variants (L_m=1, N_h=8 for X-CLIP-B while N_h=12 for X-CLIP-L). The text encoder is the same as in CLIP [10]. For Florence, we replace the cross-frame communication transformer with the pretrained CoSwin-H [13] visual encoder. We stack a 4-layer multi-frame integration transformer on top of CoSwin-H. The text encoder is the same as in Florence [13]. In both X-CLIP and X-Florence, the number of the video-specific prompting blocks is set to 2.

* Work done during internship at Microsoft Research.
† Corresponding authors: houwen.peng@microsoft.com, gfmeng@nlpr.ia.ac.cn.

Table 1: The training hyperparameters for all the experiments.

	Fully-sup.	Few-shot	Zero-shot
Optimisation			
Optimizer		AdamW	
Optimizer betas		(0.9, 0.98)	
Batch size	256	64	256
Learning rate schedule		cosine	
Linear warmup epochs		5	
Base learning rate	8e-6	2e-6	8e-6
Minimal learning rate	8e-8	2e-8	8e-8
Epochs	30	50	10
Data augmentation			
RandomFlip		0.5	
MultiScaleCrop		(1, 0.875, 0.75, 0.66)	
ColorJitter		0.8	
GrayScale		0.2	
Label smoothing		0.1	
Mixup		0.8	
Cutmix		1.0	
Other regularisation			
Weight decay		0.001	

1.2 Other compared architectures

In few-shot experiments, we implemented the Video Swin [9], TSM [8] and TimeSformer [2] using MMAction2 [5] library with the default hyperparameters. The TSM-R50 is initialized with ImageNet-1k pretraining, while the Video Swin-B and TimeSformer are initialized with ImageNet-21k pretraining.

2 Hyperparameter Details

In this section, we present the elaborated training hyperparameters in Sec. 2.1 and the hand-craft prompt templates in the Tab. 9 of the main manuscript in Sec. 2.2.

2.1 Training Hyperparameters

Tab. 1 presents the hyperparameters for our experiments, corresponding to Section 4.2-4.4 of the main manuscript. It is noteworthy that the learning rate of the randomly initialized parameters is 10× higher than the base learning rate. All the expanded models are trained with 32 NVIDIA 32G V100 GPUs.

2.2 Hand-craft Prompt Templates

In Tab. 9 of the main manuscript, we compare our video-specific prompting scheme with the existing prompt ensemble method [10] and demonstrate the superiority of our method. We construct 16 hand-craft templates totally. We randomly choose one template in each training iteration, and the result in inference is the average result of all templates. The complete list of templates is as follows: a photo of action {label}; a picture of action {label}; Human action of {label}; {label}, an action; {label}, this is an action; {label}, a video of action; Playing action of {label}; {label}; Playing a kind of action, {label}; Doing a kind of action, {label}; Look, the human is {label}; Can you recognize the action of {label}; Video classification of {label}; A video of {label}; The man is {label}; The woman is {label}.

3 Datasets and Evaluation Protocols

In this section, we overview the four datasets briefly in Sec. 3.1. Then, we provide the evaluation protocols for different experiment settings, *i.e.*, zero-shot, few-shot and fully-supervised in Sec. 3.2-3.4, respectively.

3.1 Datasets Overview

- *Kinetics-400&600.* The Kinetics [6,3] dataset consists of 10-second video clips collected from YouTube. In particular, Kinetics-400 [6] consists of ∼240k training videos and ∼20k validation videos with 400 classes, while Kinetics-600 [3] consists of ∼410k training videos and ∼29k validation videos from 600 classes.
- *UCF-101* [11]. UCF-101 is a video recognition dataset for realistic actions, collected from YouTube, including $13,320$ video clips with 101 action categories in total. There are three splits of the training and test data.
- *HMDB-51* [7]. It has around $7,000$ videos with 51 classes, which is relatively small compared to UCF-101 and Kinetics. HMDB-51 has three splits of the training and test data.

3.2 Fully-supervised Experiments

We conduct the fully-supervised experiments on Kinetics-400&600. We use the complete training and validation sets for training and inference, respectively. During training, a sparse sampling strategy [12] is used. The number of frames is set to 8 or 16. We spatially scale the shorter side of each frame to 256 and take a 224 center crop. Following [9,1,2], we adopt the multi-view inference with 3 spatial crops and 4 temporal clips.

Table 2: Comparison with different video encoders. The video encoders are adapted from ViT-B/16 [10]. The fully-supervised experiment is conducted on Kinetics-400 [6]. The few-shot(2-shot) experiments are conducted on HMDB-51, and zero-shot experiments are conducted on UCF-101 [11].

Method	Zero-shot	Few-shot	Fully-supervised	FLOPs
CLIP-One	62.5	46.2	69.9	14
CLIP-Joint	69.3	41.3	82.1	184
X-CLIP(Ours)	**70.0**(+0.7)	**50.8**(+4.6)	**82.3**(+0.2)	145

3.3 Few-shot Experiments

We randomly sample 2, 4, 8 and 16 videos from each class on UCF-101 and HMDB-51 for constructing the training set. For evaluation, we use the first split of the test set on UCF-101 and HMDB-51. We report the results with a single view of 32 frames.

3.4 Zero-shot Experiments

We train X-CLIP-B/16 with 32 frames on Kinetics-400. The single-view inference is adopted for our method. The same as [4,10], we apply the following two evaluation protocols in our zero-shot experiments. 1) *Evaluation for HMDB-51 and UCF-101.* Following [10], the prediction is conducted on the three splits of the test data, and we report the average top-1 accuracy and standard deviation. 2) *Evaluation for Kinetics-600.* Following [4], the 220 new categories outside Kinetics-400 [6] in Kinetics-600 are used for evaluation. The evaluation is conducted three times. For each iteration, we randomly sampled 160 categories for evaluation from the 220 categories in Kinetics-600.

4 Additional Experiments Analysis

In this section, we further compare different methods of adapting an image encoder to a video encoder in Sec. 4.1. Besides, we provide an analysis of aligning the ImageNet pretrained video encoder and the CLIP pretrained text encoder in Sec. 4.2. Last, we further evaluate our proposed cross-frame communication transformer and multi-frame integration transformer on a simple single-modality classification setting in Sec. 4.3.

4.1 Comparison with other video encoders adapted from images

Researchers have proposed several ways of adapting an image encoder to a video encoder [1,2]. We compare with two existing methods in Tab. 2. The first method is named "CLIP-One", in which we randomly sample one frame and feed it to the pretrained image encoder. The second method is named "CLIP-Joint",

Table 3: Comparison between the multi-modal framework and single-modal framework under ImageNet pretraining.

Method	Zero-shot	Few-shot
w/o text	/	39.4
w/ text	**62.8**	**50.7**(+11.3)

(a) ImageNet-21k pretraining.

Method	Zero-shot	Few-shot
w/o text	/	10.8
w/ text	**58.0**	**46.0**(+35.2)

(b) ImageNet-1k pretraining.

Table 4: Evaluating the proposed architecture in the single-modality framework.

Method	Top-1(%)	Top-5(%)
ViT-B/32-Mean	45.3	68.5
ViT-B/32 (Ours)	**47.8**(+2.5)	**71.8**(+3.3)

where we apply the joint space-time attention [1] that simply forwards all spatio-temporal tokens extracted from the video through the image encoder. Although the CLIP-Joint also considers global spatio-temporal information in videos, it takes more computational overhead than our proposed X-CLIP. What is more, our method surpasses the CLIP-Joint by +0.2% and +9.5% in the fully-supervised and few-shot experiments, respectively. Compared to the CLIP-One, X-CLIP is much better, indicating the efficacy of adapting an image encoder to a video encoder. We conjecture the reasons are two-fold. 1) CLIP-Joint considers the joint spatio-temporal tokens and thus breaks the customary input pattern of the pretrained image encoder, which may impede the representation ability. In contrast, our method maintains the input pattern of the pretrained image encoder via modeling frame-level information, thus leveraging the strong representation ability of the pretrained image encoder. 2) The joint space-time attention requires more training data and training time to converge than our method.

4.2 Can ImageNet pretrained video encoder align with CLIP pretrained text encoder?

We have demonstrated that the video encoder with ImageNet pretraining still achieves competitive performance on the fully-supervised experiment in Tab. 10 of the main manuscript. However, the embedding space of the ImageNet pretrained visual encoder is not well aligned with that of the CLIP pretrained text encoder. It raises a question: *can we align the two embedding spaces without the web-scale joint pretraining, and then transfer the knowledge to zero-shot experiments?* To answer this question, we build an ImageNet pretrained video encoder and extract the text encoder from the pretrained CLIP. Then, we finetune the video encoder with the text supervision on Kinetics-400 to align the two embedding spaces. As a comparison, we also finetune a same video encoder but supervised by the discrete one-hot labels. Finally, we conduct the few-shot and zero-shot experiments to

verify the transfer ability of the two models. The categories in few-shot and zero-shot experiments are not seen in finetuning. From Tab. 3, we can observe that the aligned model, *i.e.,* the model supervised by text information, achieves superior performance and surpasses the unaligned model by a large margin. It indicates that the ImageNet pretrained video encoder can still align with the CLIP pretrained text encoder by an acquired finetuning process using limited samples. The results also show the generality and flexibility of our proposed framework.

4.3 Evaluation of the proposed architectures in the single-modality framework

We further evaluate the proposed cross-frame communication transformer and multi-frame integration transformer on a simple classification setting, *i.e.,* training from scratch with a single-modality framework on Kinetics-400. We use ViT-B/32_{8f} as the backbone and adopt a fully-connected layer as the classification head. ViT-B/32-Mean averages the representation of all frames, while our method uses the cross-frame attention and stacks 1-layer multi-frame integration transformer on the top. We train both models 100 epochs with a learning rate 1×10^{-4}, and all the other hyperparameters are the same as in Tab. 1. From Tab. 4, it can be seen that our method outperforms the baseline +2.5% in terms of top-1 accuracy, which illustrates that our proposed architecture does not rely on pretraining and can help general video classification.

References

1. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: ICCV. pp. 6836–6846 (2021)
2. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML. pp. 813–824 (2021)
3. Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., Zisserman, A.: A short note about kinetics-600. arXiv preprint arXiv:1808.01340 (2018)
4. Chen, S., Huang, D.: Elaborative rehearsal for zero-shot action recognition. In: ICCV. pp. 13638–13647 (2021)
5. Contributors, M.: Openmmlab's next generation video understanding toolbox and benchmark. `https://github.com/open-mmlab/mmaction2` (2020)
6. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
7. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV. pp. 2556–2563 (2011)
8. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: ICCV. pp. 7083–7093 (2019)
9. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: CVPR (2022)
10. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
11. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
12. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: ECCV. pp. 20–36 (2016)
13. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021)