20220830

Expanding Language–Image Pretrained Model for General Video Recognition

Expanding Language-Image Pretrained Models for General Video Recognition

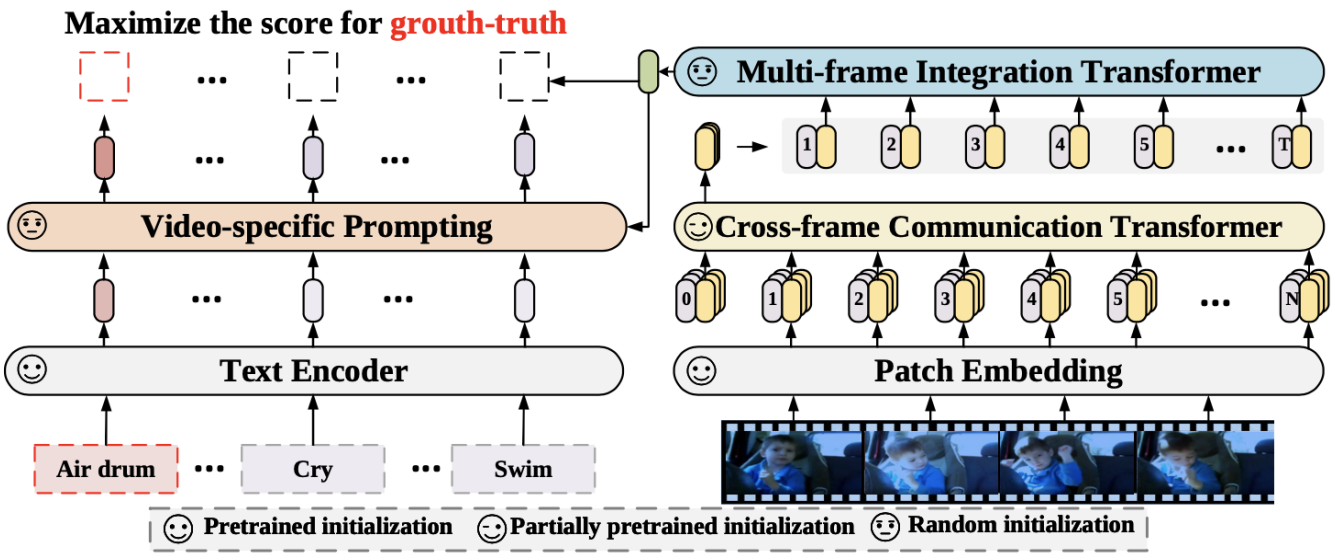
Bolin Ni[1,2 *], Houwen Peng[1 †], Minghao Chen[1,3 *], Songyang Zhang[4], Gaofeng Meng[2], Jianlong Fu[1], Shiming Xiang[2], Haibin Ling[3]

[1] Microsoft Research [2] Chinese Academy of Sciences
[3] Stony Brook University [4] University of Rochester

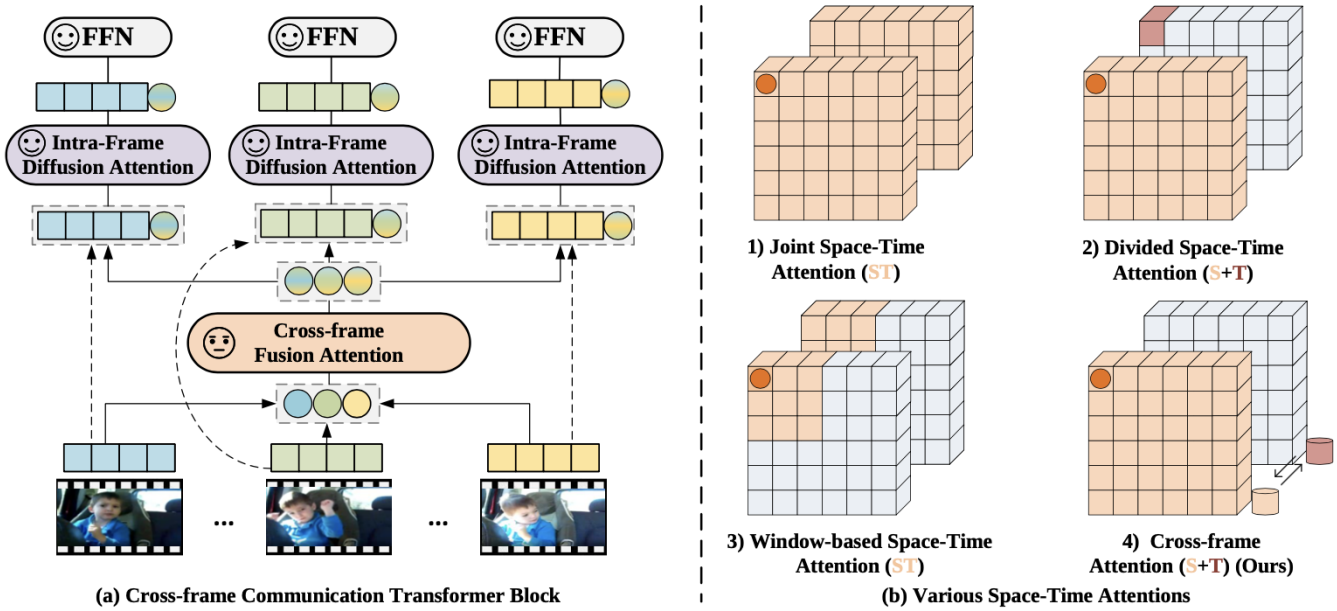
Motivation

- Task: Language–Video pretraining
- Questions:
  - 从头训练language–video的大模型浪费时间和资源，所以将Language–Image模型adapt过来
  - 如何利用视频的temporal信息：cross–frame communication attention
  - 怎样得到video的text representation：video–specific prompting，结合视觉信息

Method



Cross-frame Communication Transformer – 利用视频的时域信息



1. 对每一帧的class token做线性变换，得到message token，代表这一帧的信息
2. 在message tokens之间做attention (CFA)
3. 做完attention的message token重新concat到每一帧中，和patch embeddings做帧内的attention (IFA)

Text encoder

1. {label} 作为text description C

2. $\bar{\mathbf{c}} = \mathbf{c} + \mathrm{MHSA}(\mathbf{c}, \bar{\mathbf{z}}),$
   $\tilde{\mathbf{c}} = \bar{\mathbf{c}} + \mathrm{FFN}(\bar{\mathbf{c}}),$ $\bar{\mathbf{z}} \in \mathbb{R}^{N \times d}$ is the average of $\{\mathbf{z}_t^{(L_c)}\}_{t=1}^{T}$
3. $\hat{\mathbf{c}} = \mathbf{c} + \boldsymbol{\alpha}\tilde{\mathbf{c}}$ used for classification

$$\mathrm{sim}(\mathbf{v}, \hat{\mathbf{c}}) = \frac{\langle \mathbf{v}, \hat{\mathbf{c}} \rangle}{\|\mathbf{v}\| \, \|\hat{\mathbf{c}}\|}$$

4. goal:

Experiments

全监督

Table 1: Comparison with state-of-the-art on Kinetics-400. We report the FLOPs and throughput per view. Throughput is measured using the GitHub repository of [29] on a V100 GPU. $*$ indicates pretraining with a video-text collection.

Method	Pretrain	Frames	Top-1	Top-5	Views	FLOPs(G)	Throughput
Methods with random initialization							
MViTv1-B, 64×3 [12]	-	64	81.2	95.1	3 × 3	455	7
Methods with ImageNet pretraining							
Uniformer-B [25]	IN-1k	32	83.0	95.4	4 × 3	259	-
TimeSformer-L [5]	IN-21k	96	80.7	94.7	1 × 3	2380	3
Mformer-HR [34]	IN-21k	16	81.1	95.2	10 × 3	959	-
Swin-L [30]	IN-21k	32	83.1	95.9	4 × 3	604	6
Swin-L (384↑) [30]	IN-21k	32	84.9	96.7	10 × 5	2107	-
MViTv2-L (312↑) [27]	IN-21k	40	86.1	97.0	5 × 3	2828	-
Methods with web-scale image pretraining							
ViViT-H/16x2 [3]	JFT-300M	32	84.8	95.8	4 × 3	8316	-
TokenLearner-L/10 [40]	JFT-300M	-	85.4	96.3	4 × 3	4076	-
CoVeR [56]	JFT-3B	-	**87.2**	-	1 × 3	-	-
Methods with web-scale language-image pretraining							
ActionCLIP-B/16 [49]	CLIP-400M	32	83.8	96.2	10 × 3	563	-
A6 [21]	CLIP-400M	16	76.9	93.5	-	-	-
MTV-H [54]	WTS*	32	**89.1**	98.2	4 × 3	3705	-
X-Florence (384↑)	FLD-900M	8	86.2	96.6	4 × 3	2114	6
X-Florence	FLD-900M	32	86.5	96.9	4 × 3	2822	2
X-CLIP-B/16	IN-21k	8	81.1	94.7	4 × 3	145	33
X-CLIP-B/32	CLIP-400M	8	80.4	95.0	4 × 3	39	136
X-CLIP-B/32		16	81.1	95.5	4 × 3	75	69
X-CLIP-B/16		8	83.8	96.7	4 × 3	145	33
X-CLIP-B/16		16	84.7	96.8	4 × 3	287	17
X-CLIP-L/14		8	87.1	97.6	4 × 3	658	8
X-CLIP-L/14 (336↑)		16	**87.7**	97.4	4 × 3	3086	2

Table 2: Comparison with state-of-the-art on Kinetics-600.

Method	Pretrain	Frames	Top-1	Top-5	Views	FLOPs	Throughput
Methods with random initialization							
MViT-B-24, 32×3 [12]	-	32	83.8	96.3	5 × 1	236	-
Methods with ImageNet pretraining							
Swin-L (384↑) [30]	IN-21k	32	86.1	97.3	10 × 5	2107	-
Methods with web-scale pretraining							
ViViT-L/16x2 320 [3]	JFT-300M	32	83.0	95.7	4 × 3	3992	-
ViViT-H/16x2 [3]	JFT-300M	32	85.8	96.5	4 × 3	8316	-
TokenLearner-L/10 [40]	JFT-300M	-	86.3	97.0	4 × 3	4076	-
Florence (384↑) [55]	FLD-900M	-	87.8	97.8	4 × 3	-	-
CoVeR [56]	JFT-3B	-	**87.9**	-	1 × 3	-	-
MTV-H [54]	WTS*	32	**89.6**	98.3	4 × 3	3705	-
X-CLIP-B/16		8	85.3	97.1	4 × 3	145	74
X-CLIP-B/16	CLIP-400M	16	85.8	97.3	4 × 3	287	40
X-CLIP-L/14		8	**88.3**	97.7	4 × 3	658	20

Zero-shot: test的类别在训练时没见过

Table 3: Zero-shot performances on HMDB51 [24] and UCF101 [42].

Method	HMDB-51	UCF-101
MTE [53]	19.7 ± 1.6	15.8 ± 1.3
ASR [50]	21.8 ± 0.9	24.4 ± 1.0
ZSECOC [35]	22.6 ± 1.2	15.1 ± 1.7
UR [64]	24.4 ± 1.6	17.5 ± 1.6
TS-GCN [15]	23.2 ± 3.0	34.2 ± 3.1
E2E [6]	32.7	48
ER-ZSAR [8]	35.3 ± 4.6	51.8 ± 2.9
ActionCLIP [49]	40.8 ± 5.4	58.3 ± 3.4
X-CLIP-B/16	**44.6** ± **5.2** (+3.8)	**72.0** ± **2.3** (+13.7)
X-Florence	**48.4** ± **4.9** (+7.6)	**73.2** ± **4.2** (+14.9)

Table 4: Zero-shot performance on Kinetics-600 [7].

Method	Top-1 Acc.	Top-5 Acc.
DEVISE [14]	23.8 ± 0.3	51.0 ± 0.6
ALE [1]	23.4 ± 0.8	50.3 ± 1.4
SJE [2]	22.3 ± 0.6	48.2 ± 0.4
ESZSL [39]	22.9 ± 1.2	48.3 ± 0.8
DEM [57]	23.6 ± 0.7	49.5 ± 0.4
GCN [17]	22.3 ± 0.6	49.7 ± 0.6
ER-ZSAR [8]	42.1 ± 1.4	73.1 ± 0.3
X-CLIP-B/16	**65.2** ± **0.4** (+23.1)	**86.1** ± **0.8** (+13.0)
X-Florence	**68.8** ± **0.9** (+26.7)	**88.4** ± **0.6** (+15.3)

消融实验

Table 6: Component-wise analysis of our X-CLIP and other techniques.

Components	Top-1.(%)
Baseline(CLIP-Mean)	80.0
+ Cross-frame Communication	81.2(+1.2)
+ Multi-frame Integration	81.7(+1.7)
+ Video-specific Prompt	82.3(+2.3)
Techniques	
+ 4×3-views Inference	83.8(+3.8)

Table 7: Ablation study on which part to finetune. ✓means finetuning. The CUDA memory is calculated on 2 video inputs, each containing 8 frames.

Visual	Text	Zero.	Few.	Fully.	Mem.(G)
✓	✓	**72.9**	**54.6**	**82.4**	22
✓	✗	**70.0**	50.8	**82.3**	6
✗	✓	66.8	**53.4**	79.3	20
✗	✗	64.2	47.3	79.1	4

Table 8: Ablation study on the effect of the text information.

Method	Zero-shot	Few-shot	Fully.
w/o text	/	32.0	81.6
w/ text	**70.0**	**50.8**(+18.8)	**82.3**(+0.7)

Table 9: Comparison with different prompting methods.

Method	Fully.	Few.	Zero.
w/o prompt	81.7	49.6	63.2
Ensemble. [37]	81.7	49.6	63.9
Vectors. [61]	82.0	49.9	63.2
Ours	**82.3**(+0.3)	**50.8**(+0.9)	**70.0**(+6.1)