AIM: Adapting Image Models for Efficient Video Action Recognition

Taojiannan Yang[1], Yi Zhu[2], Yusheng Xie[2], Aston Zhang[2], Chen Chen[1] , Mu Li[2]
[1]University of Central Florida
[2]Amazon Web Services

ICLR 2023

TA: Geonhee Han (rtrt505@korea.ac.kr), Sehwan Park (shp216@korea.ac.kr)
2025. 8. 8.

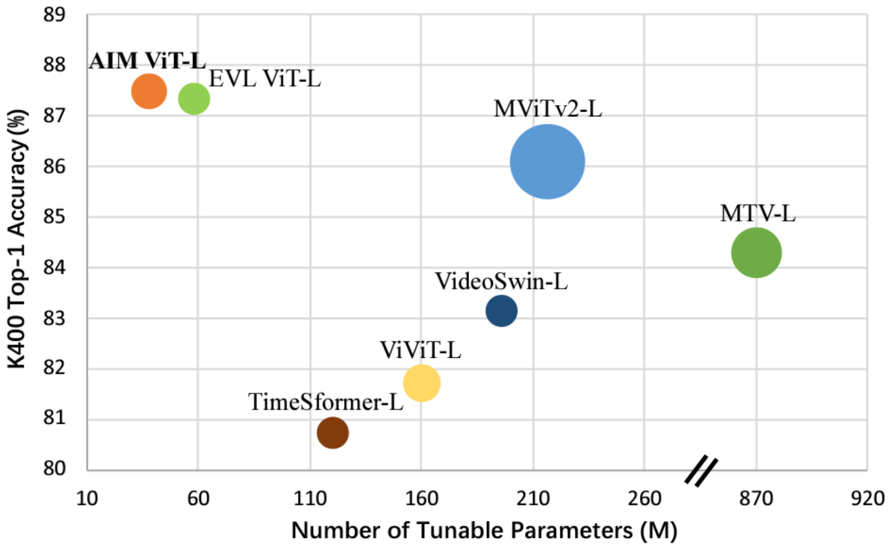
KOREA UNIVERSITY

MIIL Multimodal Interactive Intelligence Laboratory

Overview

Overview

- Training video models is drastically more expensive in both computation resource and time than image models.
- This paper introduces a new efficient image-to-video transfer method, dubbed AIM.
- AIM is effective and efficient in terms of #parameter, #data, time, and memory footprint.
- This paper might be useful for researchers who:
  - want to train large-scale video models in Lab.
  - trying to use CLIP-pretrained ViT backbone and finetune it on downstream tasks efficiently.

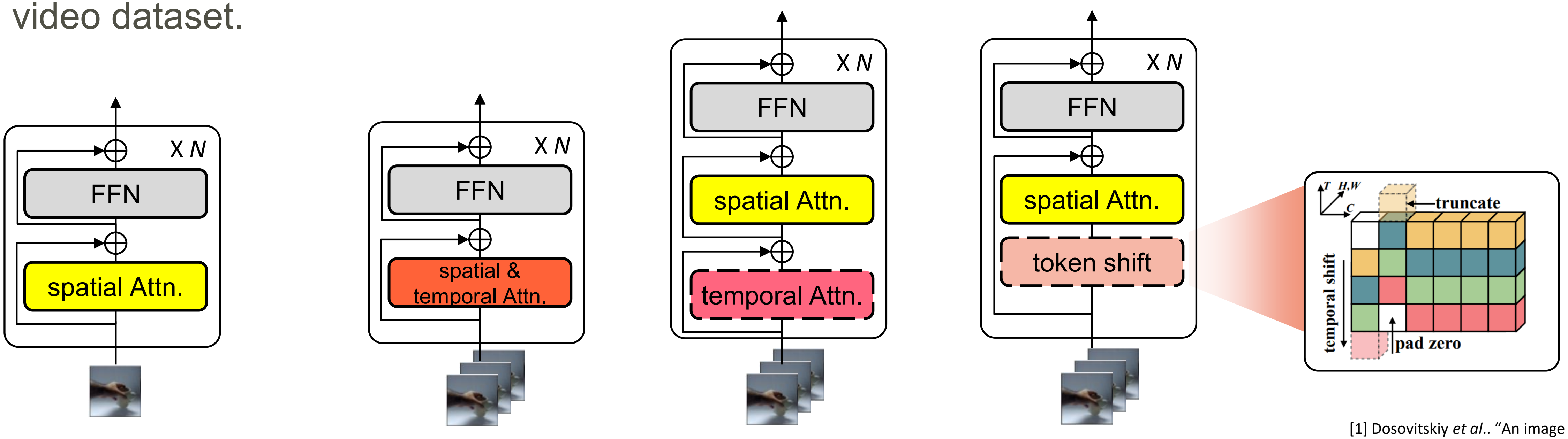


KOREA UNIVERSITY Multimodal Interactive Intelligence Laboratory

Introduction & Related work

Transferring image models to video models

- Video models heavily rely on image-pretrained models due to lack of training video data and large model capacity.
- Image-to-video transfer often requires to modify the image models and full-finetune the models on video dataset.



(a) spatial ViT [1]

- $\mathcal{O}(H^2W^2)$

(b) joint attention [2, 3]

- $\mathcal{O}(T^2H^2W^2)$
- Full spatio-temporal Attn.
- Expensive

(c) factorized attention [3]

- $\mathcal{O}(TH^2W^2 + T^2HW)$
- More efficient than (c).
- Additional parameters.
- Limited temporal modeling.

(d) token shift [4]

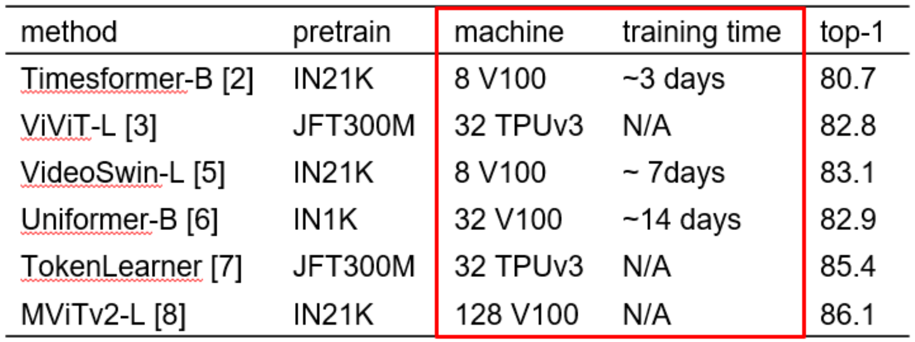
- $\mathcal{O}(TH^2W^2)$
- Approximation of (c).
- More efficient than (c) and (d).
- Limited temporal modeling.

[1] Dosovitskiy *et al*.. "An image is worth 16x16 words: Transformers for image recognition at scale." *ICLR*. 2021.
[2] Arnab *et al*.. "ViViT: A video video transformer." *ICCV*. 2021.
[3] Bertasius *et al*.. "Is space-time attention all you need for video understanding?" *ICML*. 2021
[4] Bulat *et al*.. "Space-time mixing attention for video transformer." *NeurIPS*. 2021.

Introduction & Related work

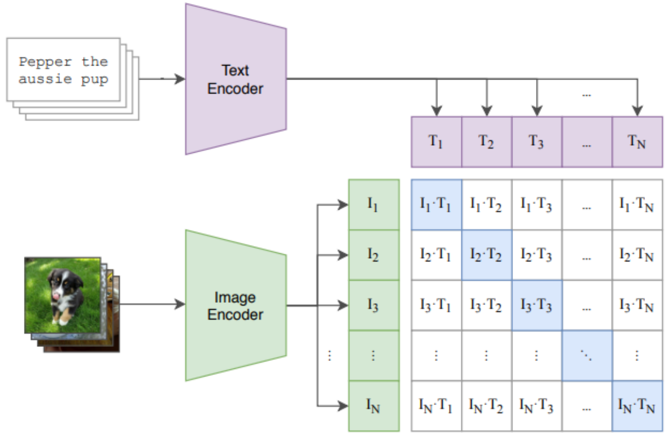
Expensive image-to-video transfer hinders the use of foundation models

- Most of video model should be fully finetuned on downstream video benchmarks using image-pretrained weights.
- However, full-finetuning is expensive.
- Given a generalizable image encoder, it would be more efficient to preserve such good representations. [9, 10, 11].



method	pretrain	machine	training time	top-1
Timesformer-B [2]	IN21K	8 V100	~3 days	80.7
ViViT-L [3]	JFT300M	32 TPUv3	N/A	82.8
VideoSwin-L [5]	IN21K	8 V100	~ 7days	83.1
Uniformer-B [6]	IN1K	32 V100	~14 days	82.9
TokenLearner [7]	JFT300M	32 TPUv3	N/A	85.4
MViTv2-L [8]	IN21K	128 V100	N/A	86.1

<Comparison of SOTA models on Kinetics-400>



	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

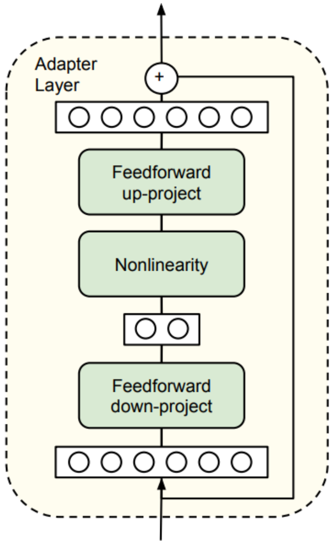
<CLIP [9]>

[2] Arnab *et al*.. “ViViT: A video video transformer.” *ICCV*. 2021.
[3] Bertasius *et al*.. “Is space-time attention all you need for video understanding?” *ICML*. 2021.
[5] Liu *et al*.. “Video swin transformer.” *CVPR*. 2022.
[6] Li et al.. “Uniformer: Unified transformer for efficient spatiotemporal representation learning.” ICLR. 2022.
[7] Ryoo *et al*.. “TokenLearner: Adaptive space-time tokenization for videos.” NeurIPS. 2021.
[8] Li *et al*.. “MViTv2: Improved multiscale vision transformers for classification and detection.” *CVPR*. 2022.
[9] Radford *et al*.. “Learning transferable visual models from natural language supervision.” *ICML*. 2021.
[10] Singh *et al*.. “Revisiting weakly supervised pre-training of visual perception models.” *CVPR*. 2022.
[11] Yu *et al*.. “CoCa: Contrastive captioners are image-text foundation models.” *TMLR*. 2022.

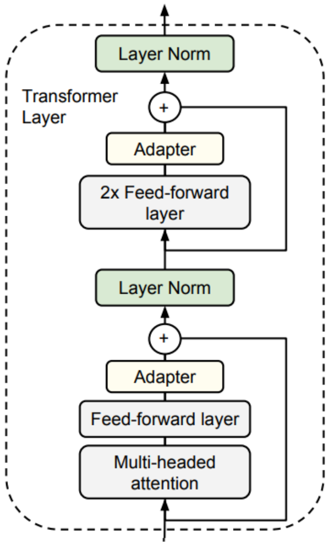
Introduction & Related work

Efficient transfer of large language models (LLMs)

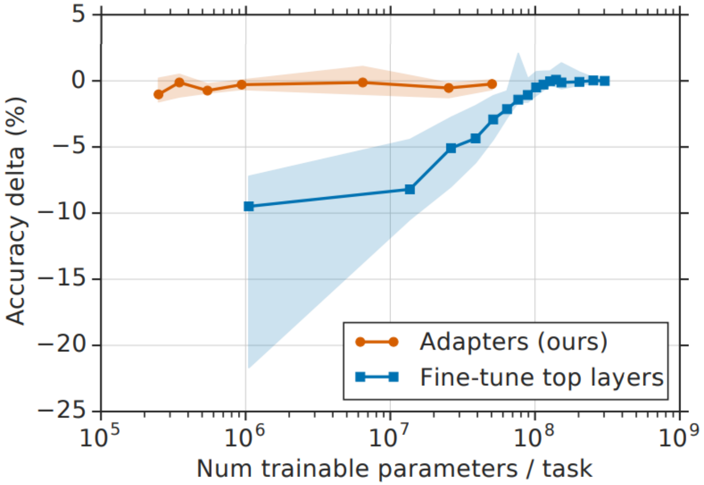
- Since LLMs [12, 13] are too large to be fully finetuned on downstream datasets, there exist lines of research streams, e.g., adapter tuning [14, 15] or prompt learning [16], try to transfer LLMs to downstream tasks efficiently.
- Adapter [14], which is the most relevant to this paper, proposes to insert lightweight neural blocks into transformer blocks and train the newly added blocks only freezing the original LLM weights.



<Adapter>



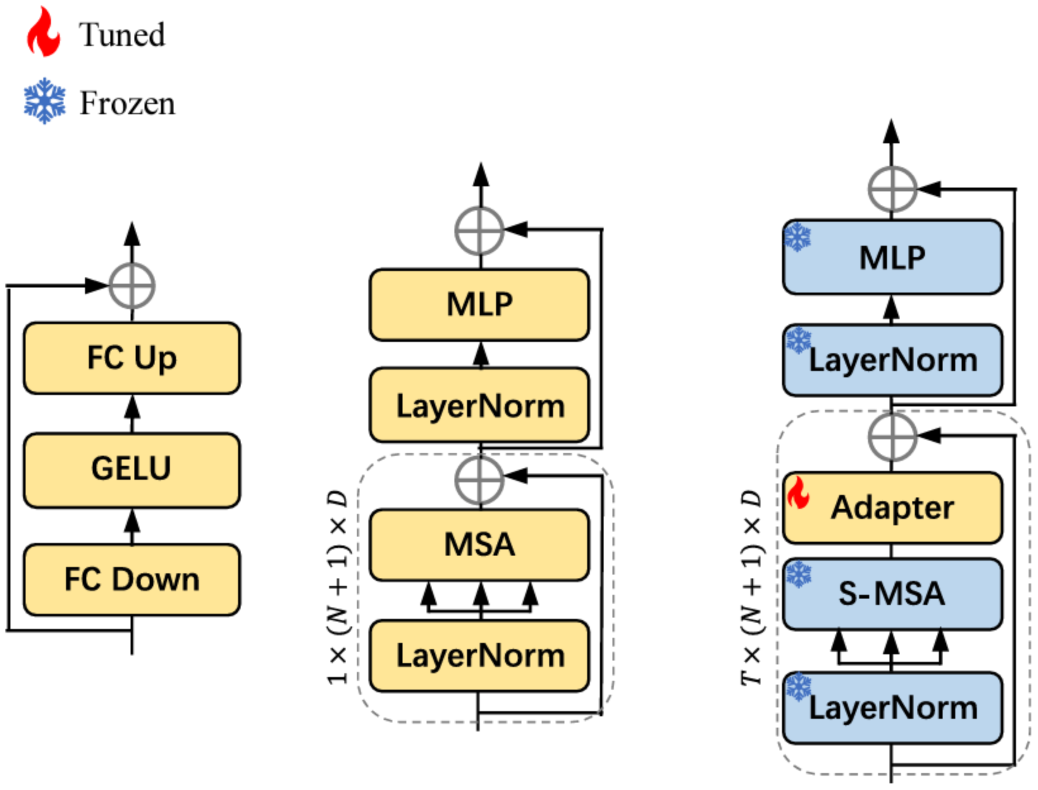
<LM w/ Adapter>



<# trainable parameters vs. performance on GLUE>

[12] Devlin et al.. “BERT: Pre-training of deep bidirectional transformers for language understanding.” *ACL*. 2019.
[13] Brown et al.. “Language models are few-shot learners.” *NeurIPS*. 2020.
[14] Houlsby *et al*.. “Parameter-efficient transfer learning for NLP.” *PMLR*. 2021.
[15] Li *et al*.. “Prefix-tuning: Optimizing continuous prompts for generation.” *ACL*. 2021.
[16] Hu *et al*.. “LoRA: Low-rank adaptation of large language models.” *ICLR*. 2022.

Method

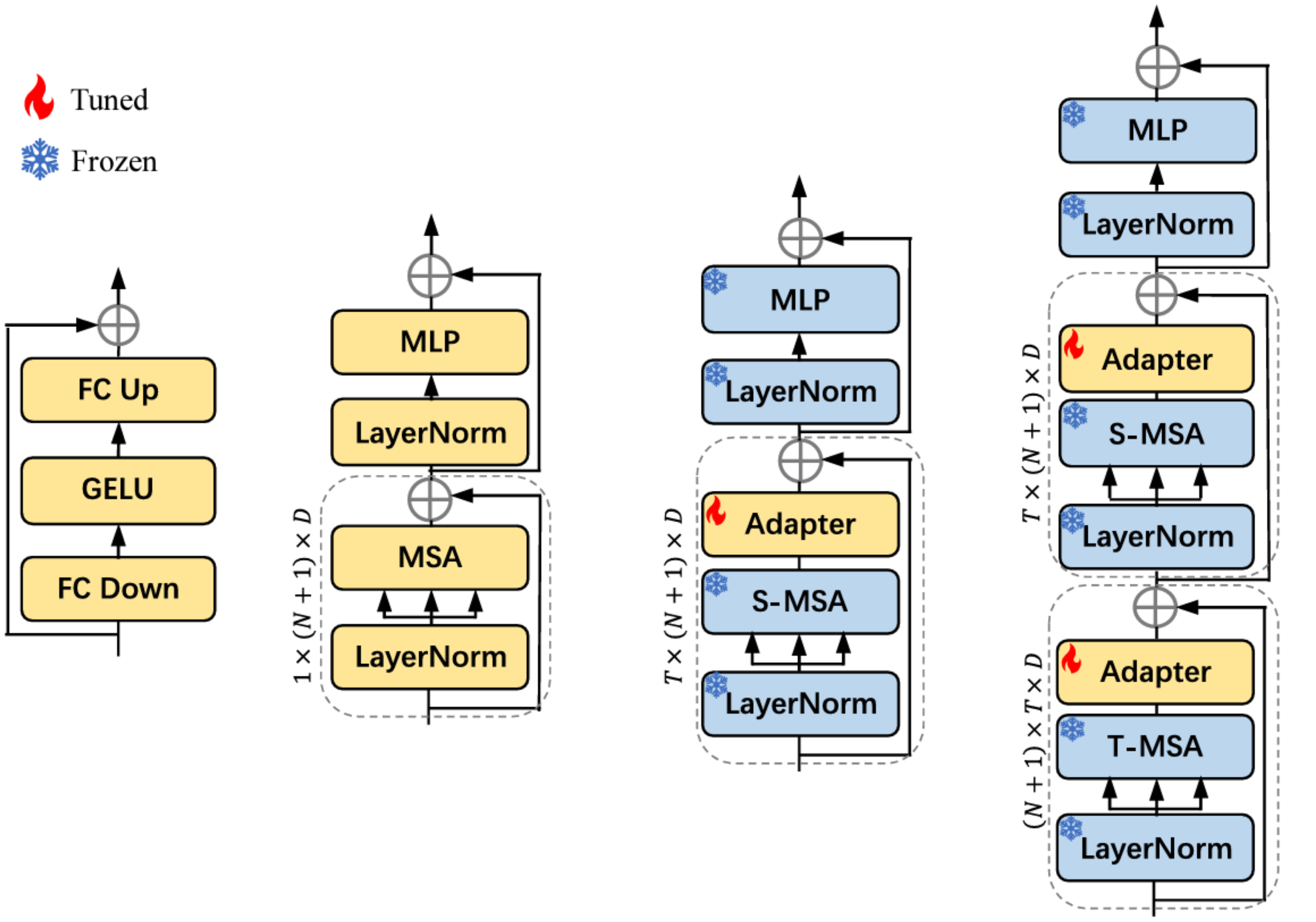


(a) Adapter

(b) ViT Block

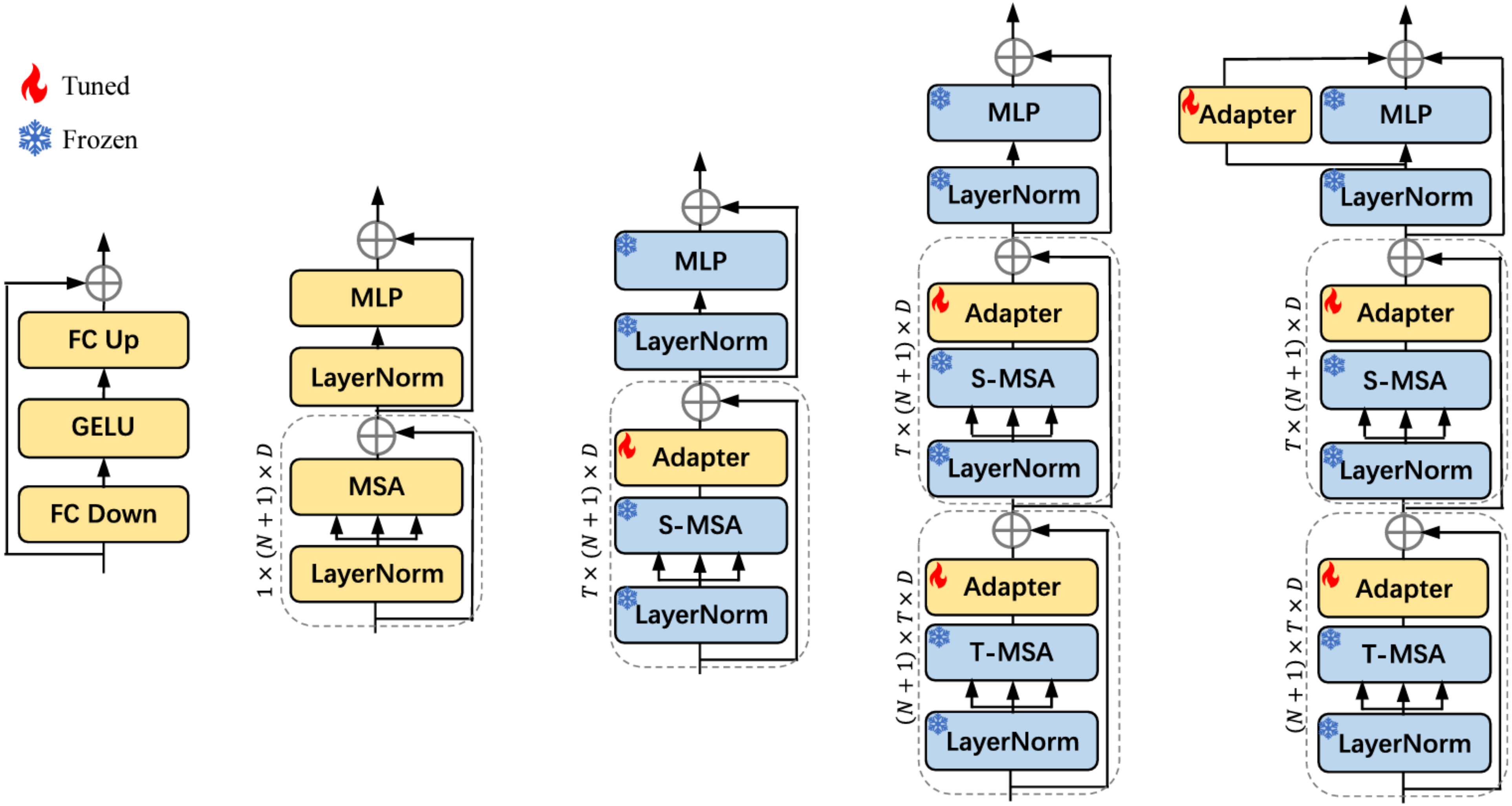
(c) Spatial Adaptation

Method



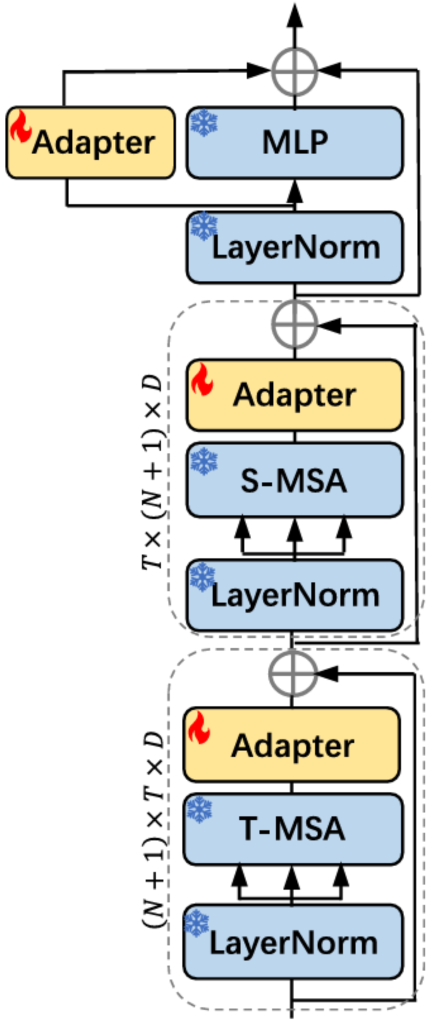
(a) Adapter (b) ViT Block (c) Spatial Adaptation (d) Temporal Adaptation

Method



(a) Adapter (b) ViT Block (c) Spatial Adaptation (d) Temporal Adaptation (e) Joint Adaptation

Method



(e) Joint Adaptation

$$z_l^T = z_{l-1} + \mathrm{Adapter}(\mathrm{T\text{-}MSA}(\mathrm{LN}(z_{l-1})))$$

$$z_l^S = z_l^T + \mathrm{Adapter}(\mathrm{S\text{-}MSA}(\mathrm{LN}(z_l^T)))$$

$$z_l = z_l^S + \mathrm{MLP}(\mathrm{LN}(z_l^S)) + s \cdot \mathrm{Adapter}(\mathrm{LN}(z_l^S))$$

Experiments

Experimental setup: datasets & pretrained models

- Datasets:

  1. Kinetics-400

     400 classes, 300k videos, 25fps. 10s per video. Appearance-oriented.

  2. Something-Something V2

     174 classes, 220k videos. 12fps. 4s per video. Motion-oriented.

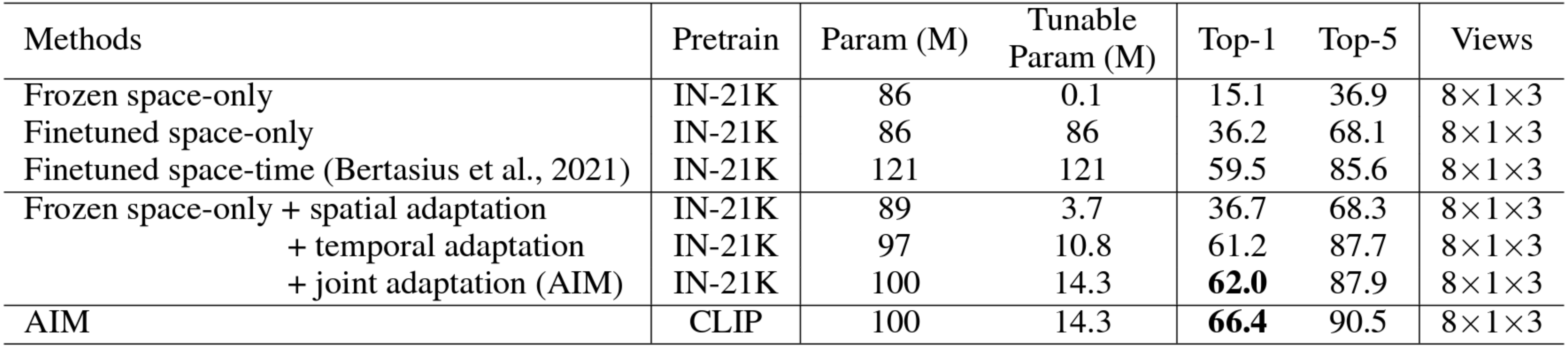
- Pretrained models:
  - CLIP-pretrained ViT-B or ViT-L

label: pull ups

label: removing sth., revealing sth. behind

Experiments

Effects of parameter-efficient transfer learning



Methods	Pretrain	Param (M)	Tunable Param (M)	Top-1	Top-5	Views
Frozen space-only	IN-21K	86	0.1	15.1	36.9	8×1×3
Finetuned space-only	IN-21K	86	86	36.2	68.1	8×1×3
Finetuned space-time (Bertasius et al., 2021)	IN-21K	121	121	59.5	85.6	8×1×3
Frozen space-only + spatial adaptation	IN-21K	89	3.7	36.7	68.3	8×1×3
+ temporal adaptation	IN-21K	97	10.8	61.2	87.7	8×1×3
+ joint adaptation (AIM)	IN-21K	100	14.3	**62.0**	87.9	8×1×3
AIM	CLIP	100	14.3	**66.4**	90.5	8×1×3

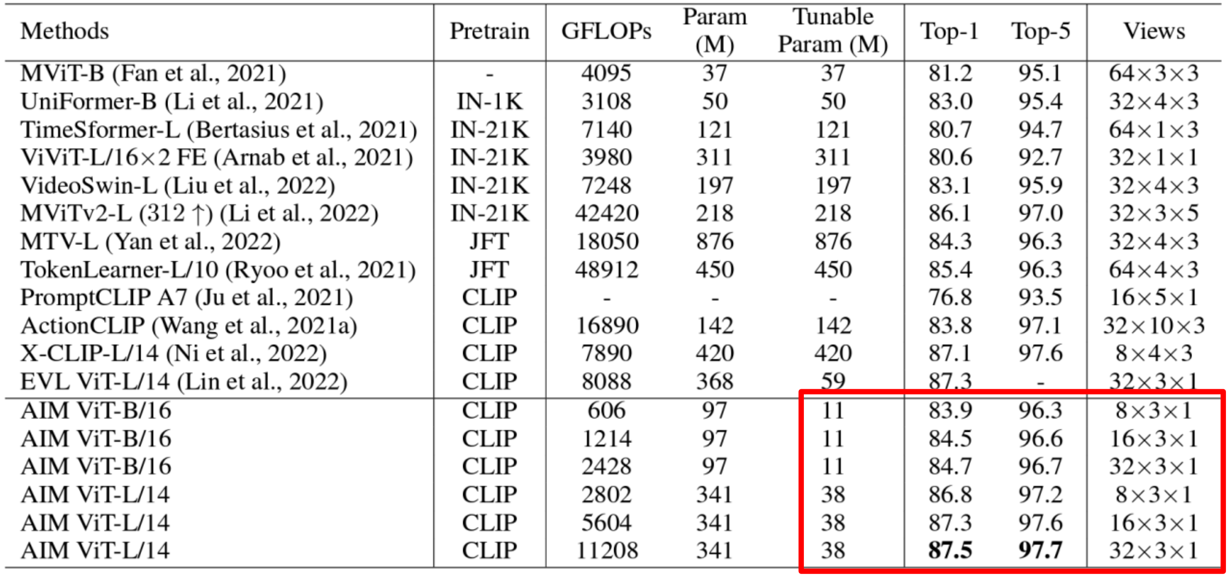
Ablation studies on Something-Something V2

Experiments

SOTA comparison on Kinetics-400 and Something-Something V2,

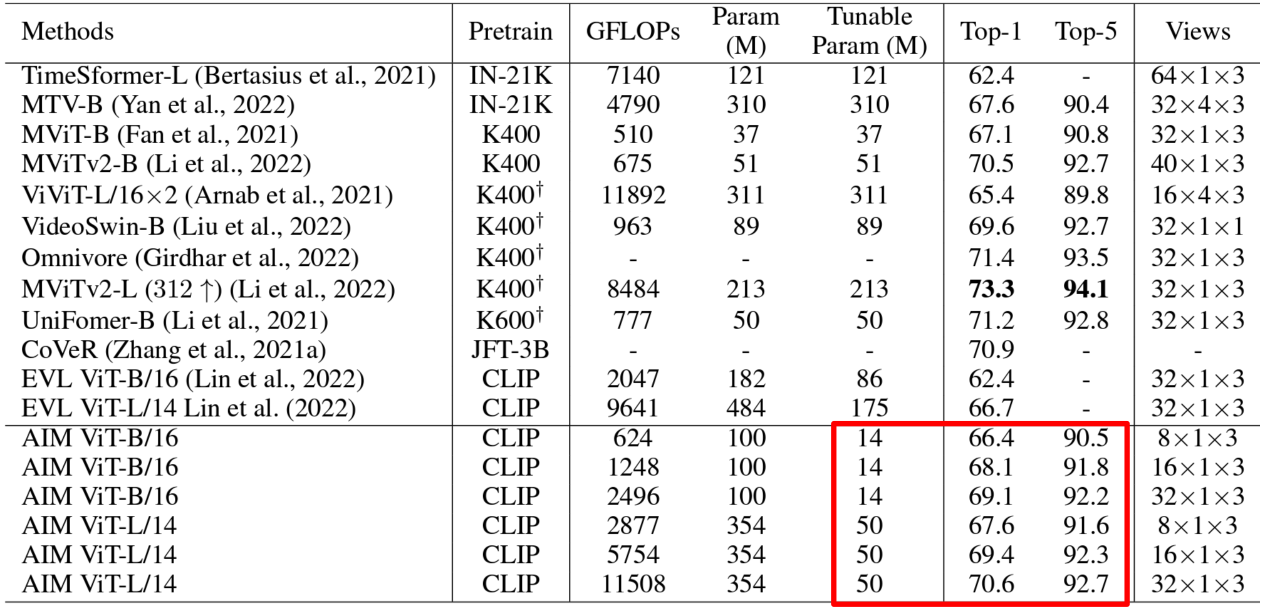
- ViTs trained with AIM achieve strong performances on both benchmarks only training ~15% of the parameters of the original ViT.

Results on Kinetics-400



Methods	Pretrain	GFLOPs	Param (M)	Tunable Param (M)	Top-1	Top-5	Views
MViT-B (Fan et al., 2021)	-	4095	37	37	81.2	95.1	64×3×3
UniFormer-B (Li et al., 2021)	IN-1K	3108	50	50	83.0	95.4	32×4×3
TimeSformer-L (Bertasius et al., 2021)	IN-21K	7140	121	121	80.7	94.7	64×1×3
ViViT-L/16×2 FE (Arnab et al., 2021)	IN-21K	3980	311	311	80.6	92.7	32×1×1
VideoSwin-L (Liu et al., 2022)	IN-21K	7248	197	197	83.1	95.9	32×4×3
MViTv2-L (312 ↑) (Li et al., 2022)	IN-21K	42420	218	218	86.1	97.0	32×3×5
MTV-L (Yan et al., 2022)	JFT	18050	876	876	84.3	96.3	32×4×3
TokenLearner-L/10 (Ryoo et al., 2021)	JFT	48912	450	450	85.4	96.3	64×4×3
PromptCLIP A7 (Ju et al., 2021)	CLIP	-	-	-	76.8	93.5	16×5×1
ActionCLIP (Wang et al., 2021a)	CLIP	16890	142	142	83.8	97.1	32×10×3
X-CLIP-L/14 (Ni et al., 2022)	CLIP	7890	420	420	87.1	97.6	8×4×3
EVL ViT-L/14 (Lin et al., 2022)	CLIP	8088	368	59	87.3	-	32×3×1
AIM ViT-B/16	CLIP	606	97	11	83.9	96.3	8×3×1
AIM ViT-B/16	CLIP	1214	97	11	84.5	96.6	16×3×1
AIM ViT-B/16	CLIP	2428	97	11	84.7	96.7	32×3×1
AIM ViT-L/14	CLIP	2802	341	38	86.8	97.2	8×3×1
AIM ViT-L/14	CLIP	5604	341	38	87.3	97.6	16×3×1
AIM ViT-L/14	CLIP	11208	341	38	**87.5**	**97.7**	32×3×1

Results on Something-Something V2

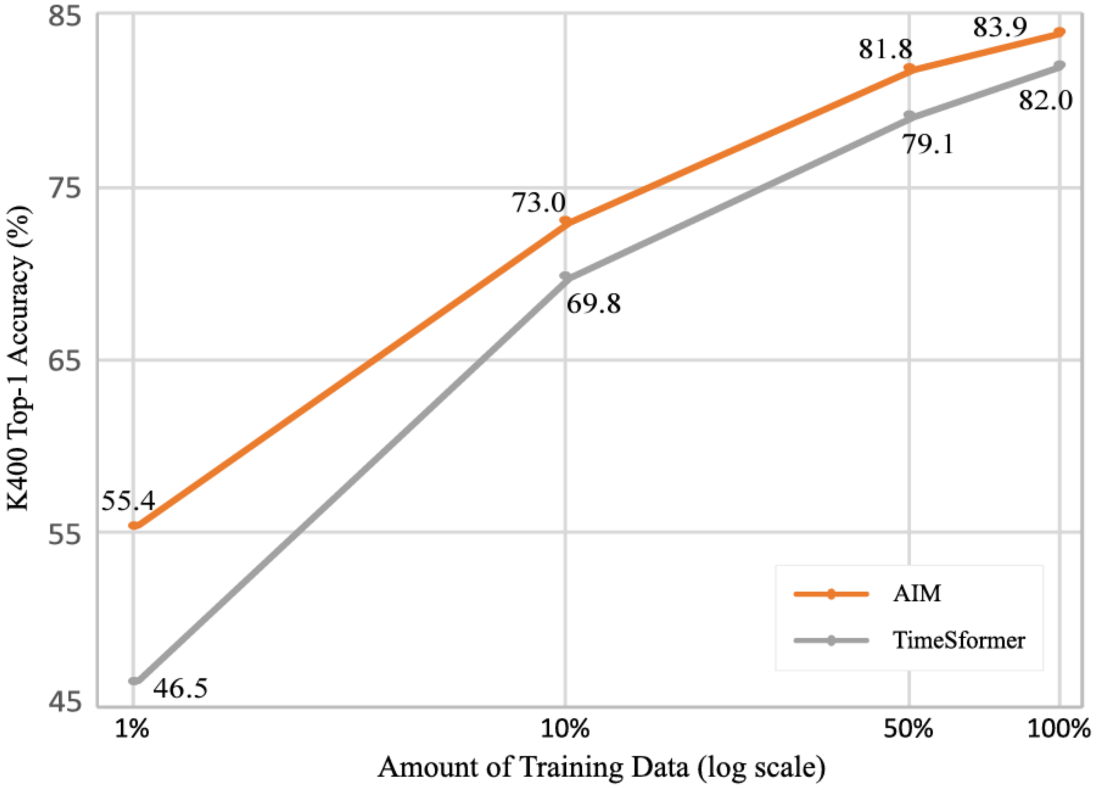


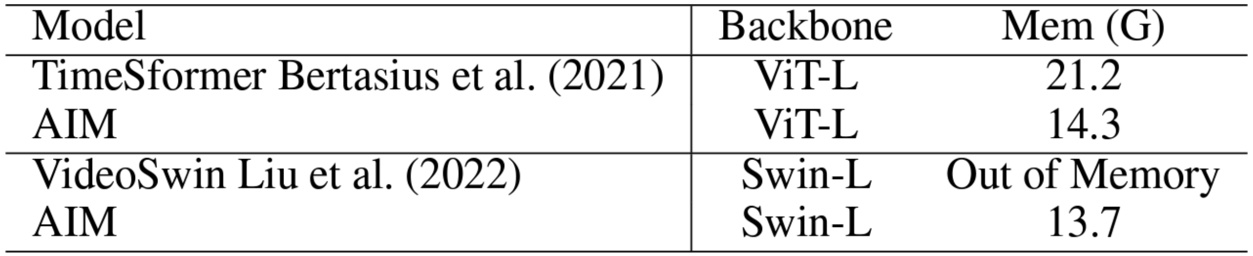
Methods	Pretrain	GFLOPs	Param (M)	Tunable Param (M)	Top-1	Top-5	Views
TimeSformer-L (Bertasius et al., 2021)	IN-21K	7140	121	121	62.4	-	64×1×3
MTV-B (Yan et al., 2022)	IN-21K	4790	310	310	67.6	90.4	32×4×3
MViT-B (Fan et al., 2021)	K400	510	37	37	67.1	90.8	32×1×3
MViTv2-B (Li et al., 2022)	K400	675	51	51	70.5	92.7	40×1×3
ViViT-L/16×2 (Arnab et al., 2021)	K400†	11892	311	311	65.4	89.8	16×4×3
VideoSwin-B (Liu et al., 2022)	K400†	963	89	89	69.6	92.7	32×1×1
Omnivore (Girdhar et al., 2022)	K400†	-	-	-	71.4	93.5	32×1×3
MViTv2-L (312 ↑) (Li et al., 2022)	K400†	8484	213	213	**73.3**	**94.1**	32×1×3
UniFomer-B (Li et al., 2021)	K600†	777	50	50	71.2	92.8	32×1×3
CoVeR (Zhang et al., 2021a)	JFT-3B	-	-	-	70.9	-	-
EVL ViT-B/16 (Lin et al., 2022)	CLIP	2047	182	86	62.4	-	32×1×3
EVL ViT-L/14 Lin et al. (2022)	CLIP	9641	484	175	66.7	-	32×1×3
AIM ViT-B/16	CLIP	624	100	14	66.4	90.5	8×1×3
AIM ViT-B/16	CLIP	1248	100	14	68.1	91.8	16×1×3
AIM ViT-B/16	CLIP	2496	100	14	69.1	92.2	32×1×3
AIM ViT-L/14	CLIP	2877	354	50	67.6	91.6	8×1×3
AIM ViT-L/14	CLIP	5754	354	50	69.4	92.3	16×1×3
AIM ViT-L/14	CLIP	11508	354	50	70.6	92.7	32×1×3

Experiments

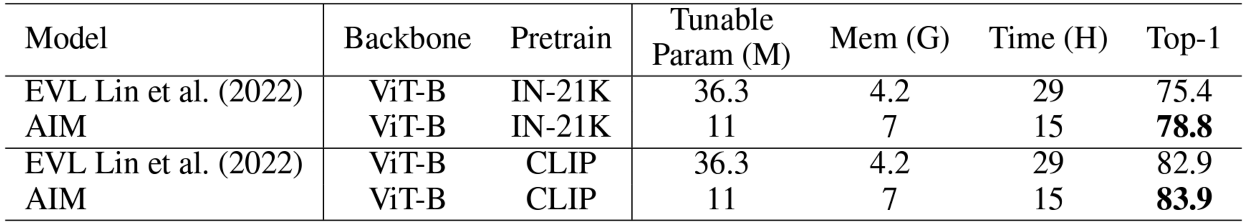
Efficiency comparison

- AIM is efficient in terms of # parameters, training time, # data, and memory.





Model	Backbone	Mem (G)
TimeSformer Bertasius et al. (2021)	ViT-L	21.2
AIM	ViT-L	14.3
VideoSwin Liu et al. (2022)	Swin-L	Out of Memory
AIM	Swin-L	13.7



Model	Backbone	Pretrain	Tunable Param (M)	Mem (G)	Time (H)	Top-1
EVL Lin et al. (2022)	ViT-B	IN-21K	36.3	4.2	29	75.4
AIM	ViT-B	IN-21K	11	7	15	**78.8**
EVL Lin et al. (2022)	ViT-B	CLIP	36.3	4.2	29	82.9
AIM	ViT-B	CLIP	11	7	15	**83.9**

Practice

Practice

Let’s practice!!!

https://colab.research.google.com/drive/1-nUSyGfRGyBYlWx04AATDQqRt82mixHy?usp=sharing