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

Presented By:

Annie, Shoubin, Nick, Junjie

Finetuning Methods for Video Understanding

In recent years, Video Understanding has typically used two techniques:

- Image Model + Temporal Module: Adding a specialized network to specifically integrate temporal reasoning
  - Think… TimeSFormer [10]
- Inflating an Existing Image Model: Stacking and connecting layers of existing image models for each frame of a given video
  - Think… Quo Vadis [3]

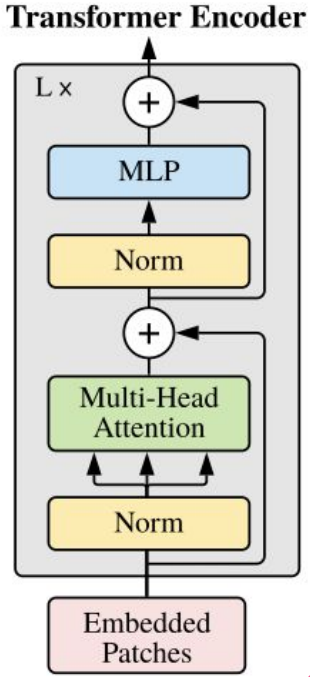
The Issue?

These traditional finetuning techniques often require either training from scratch or full finetuning of a pretrained network

- Both are costly…
  - Bertasius et Al. (2021) required tuning of 121M for Divided Space-Time TimeSFormer Model
  - Liu et Al. (2022) noted their inflation training technique required 1200 V100 GPU Hours for training VideoSwin
- Plus, poorly finetuning pretrained parameters might diminish or destroy generalized representations

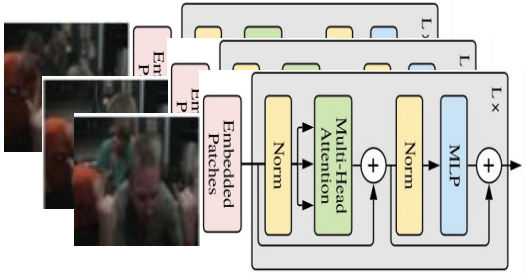
The Vision Transformer (ViT)

- Vision Transformers (ViT) have become widely adopted for CV tasks, including but not limited to Video Action Recognition
- High level Idea: Tokenize and imbed image patches, then feed through Multi-Headed Self Attention and MLP Layers



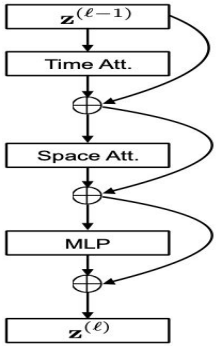
The Vision Transformer (ViT) for Video

Baseline Space-Only Approach:
Compute each frame and average the results



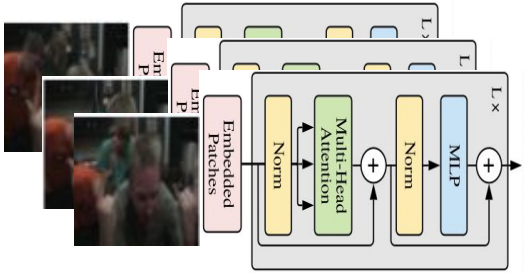
Headbanging

Space-Time Approaches:
Add new Temporal Modual to Image Model



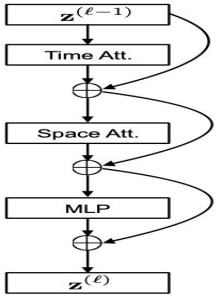
The Vision Transformer (ViT) for Video

Baseline Space-Only Approach: While simple, being time-agnostic leaves much to be desired



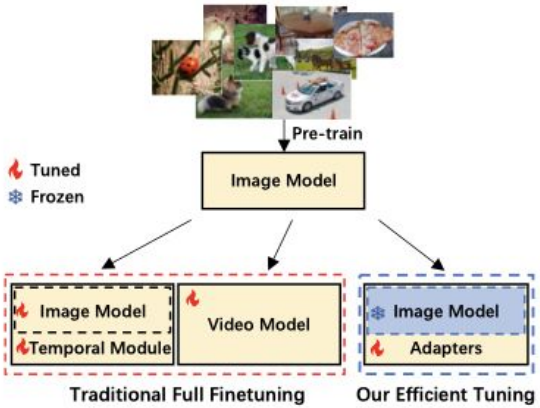
Headbanging

Space-Time Approaches: Approaches SOTA Results, but high training costs makes these models cost-probibitive



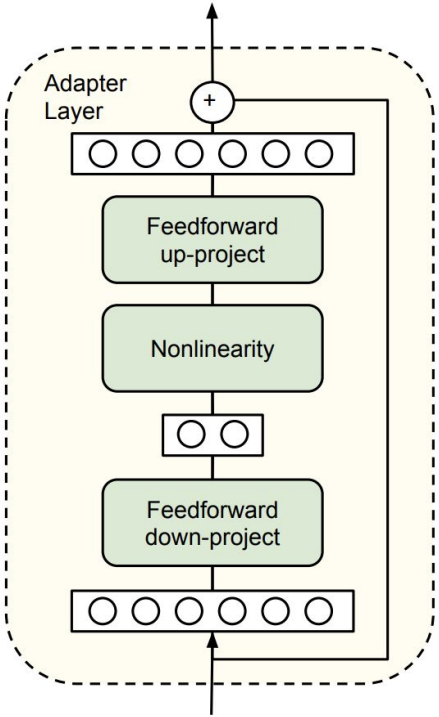
How can we *Efficiently* Tune a ViT based video model?

- Taking Inspiration from NLP… Is it possible to finetune a small batch of extra parameters (Adapters) while freezing pretrained layers?
- Advantages:
  - Training can be more efficient (tuning 38M vs 88M Parameters)
  - Can still leverage high performance of larger pretrained networks, without all of the computational overhead



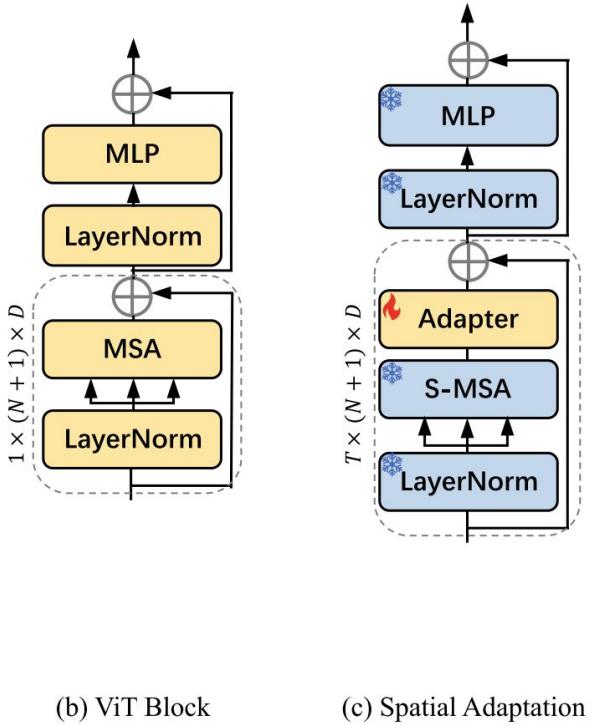
Method

Add simple Adapter module:



Spatial adaption

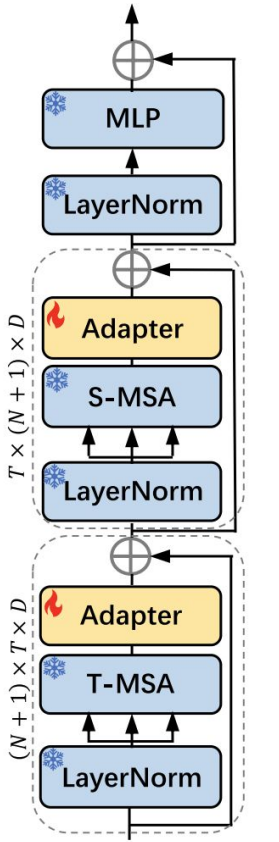
- Freeze ViT weights
- Add adapter after self-attention layer
- comparable performance with full finetuned space-only baseline



(b) ViT Block (c) Spatial Adaptation

Temporal adaptation

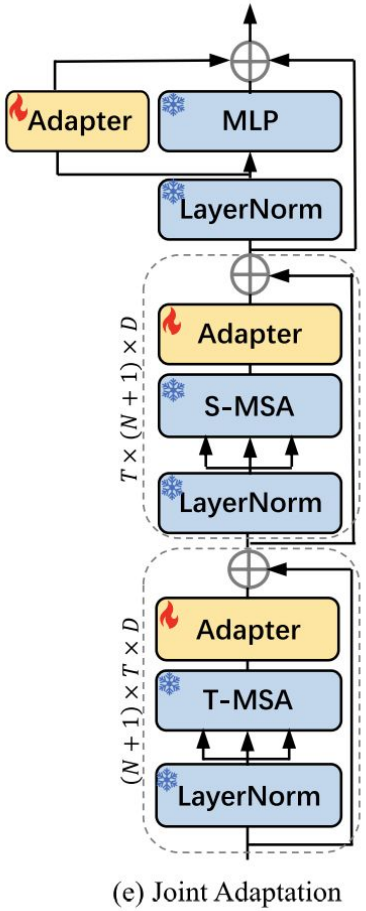
- Spatial information not enough
- Adding temporal modules too expensive
- Solution: ***reuse* the pre-trained self-attention layer in the image model**
  - Reshape input to learn relationships across time



(d) Temporal Adaptation

Final modified Transformer block

- Add final adapter to jointly tune the representations for spatiotemporal reasoning
- Final prediction: feed the average of the [class] tokens of each input frame to classification head



(e) Joint Adaptation

Experiments

Datasets:

- Kinetics-400
- Kinetics-700
- Something-Something V2
- Diving-48 (15.9k training videos. Designed to be unbiased towards static representations)

Effectiveness of proposed components

Compared the method to three baselines on Something-Something-V2

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

Effectiveness of proposed components

Spatial adaptation, temporal adaptation and joint adaptation gradually add spatiotemporal reasoning to the frozen image model

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

Effectiveness of proposed components

The approach could easily benefit from stronger image foundation models in the future

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

Comparison with SoTAs

Comparisons 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

Comparison with SoTAs

Comparisons on Kinetics-700

Method	Pretrain	Tunable Param	Top-1
VidTR-L (Zhang et al., 2021b)	IN-21K	91	70.2
MTV-L (Yan et al., 2022)	IN-21K	876	75.2
MViTv2-B (Li et al., 2022)	-	51	76.6
MViTv2-L ($40 \times 312 \uparrow$) (Li et al., 2022)	IN-21K	218	79.4
MaskFeat ($40 \times 312 \uparrow$) (Wei et al., 2022)	K700	218	**80.4**
AIM ViT-B/16	CLIP	11	76.9
AIM ViT-L/14	CLIP	38	**80.4**

Comparisons on Diving-48

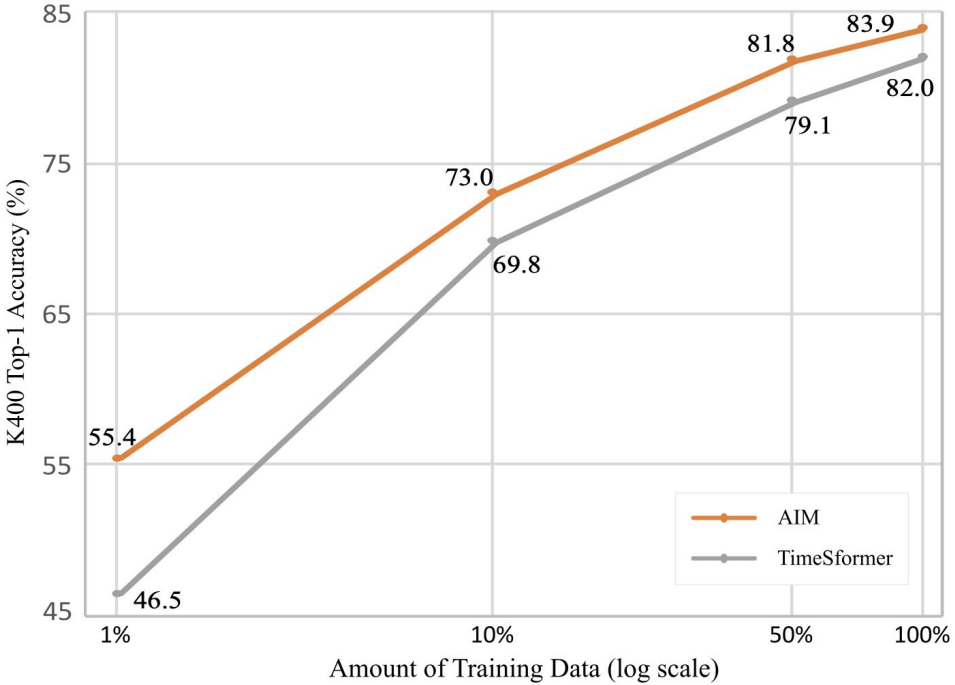
Method	Pretrain	Tunable Param	Top-1
TimeSformer-L (Bertasius et al., 2021)	IN-21K	121	81.0
VideoSwin-B (Liu et al., 2022)	IN-21K	88	81.9
BEVT (Wang et al., 2022a)	K400†	88	86.7
SIFAR-B-14 (Fan et al., 2022)	IN-21K	87	87.3
ORViT (Herzig et al., 2022)	IN-21K	160	88.0
AIM ViT-B/16	CLIP	11	88.9
AIM ViT-L/14	CLIP	38	**90.6**

Different Pre-trained Models

Different pre-trained models on Kinetics-400

Model	Backbone	Pretrain	Tunable Param (M)	Mem (G)	Time (H)	Top-1
TimeSformer	ViT-B	IN-21K	121	10	20	78.5
AIM	ViT-B	IN-21K	11	7	15	78.8
TimeSformer	ViT-B	CLIP	121	10	20	82.0
AIM	ViT-B	CLIP	11	7	15	83.9
VideoSwin-B	Swin-B	IN-21K	88	18	64	82.7
AIM	Swin-B	IN-21K	9.2	9	37	82.1

Data Efficiency



- Keep the well pre-trained image representations intact
- Have an advantage when downstream data is insufficient

Comparison: AIM v.s. PromptCLIP

AIM 😍

👁️ image model for visual tasks, flexiable to a large number of visual models

🤩 wow! reusing spatial layers as temporal layers can work?!

👁️ simple, efficient, and **unified,** more likely to be a new foundation video modeling scheme

👍 Better result but less input view, less/similar parameter!

PromptCLIP 🤔

💬 image-language model for visual tasks, need an extra text model

🤖 everything logical and make sense, but less excited

🤦‍♂️ A + B + C

😐 cool performance on a wide range tasks

Comparison: AIM v.s. PromptCLIP

Table 2: Comparison to state-of-the-art on Kinetics-400. Views = #frames × #temporal × #spatial.

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

Comparison: AIM v.s. PromptCLIP

PromptCLIP

For all the following action recognition experiments, we inherit the best practice from the ablation studies, *i.e.* prepend / append 16 prompt vectors to category names, and only use two Transformer layers (5M parameters) for temporal modeling, for its best trade-off on performance and computational cost.

AIM

Table 7: Effect of position of Adapters. Skip means adding Adapters every two blocks.

Position	Tunable Param (M)	Top-1
Bottom 6	5.6	80.7
Top 6	5.6	83.3
Skip	5.6	83.2
All	11	**83.9**