TempMe: Video Temporal Token Merging for Efficient Text-Video Retrieval

Leqi Shen[1,2,3*] **Tianxiang Hao**[1,2*] **Tao He**[5,6] **Sicheng Zhao**[2†]
Yifeng Zhang[4] **Pengzhang Liu**[4] **Yongjun Bao**[4] **Guiguang Ding**[1,2†]
[1] School of Software, Tsinghua University [2] BNRist, Tsinghua University
[3] Hangzhou Zhuoxi Institute of Brain and Intelligence
[4] JD.com [5] GRG Banking Equipment Co., Ltd. [6] South China University of Technology

Background

Text-**V**ideo **R**etrieval.

- Matching videos that correspond to specific query texts or vice versa.
- Recent studies focus on full fine-tuning of CLIP for TVR.

Limitations.

- Introducing cumbersome modules to extract video features.
- Slow inference speed severely limits their real-world applications.
- The training process of CLIP4Clip with CLIP-ViT-B/16 requires 70.1GB GPU memory usage and takes 6.5 hours.

In this work, we focus on efficient fine-tuning TVR.

Background

Challenges in efficient adaptation for TVR.

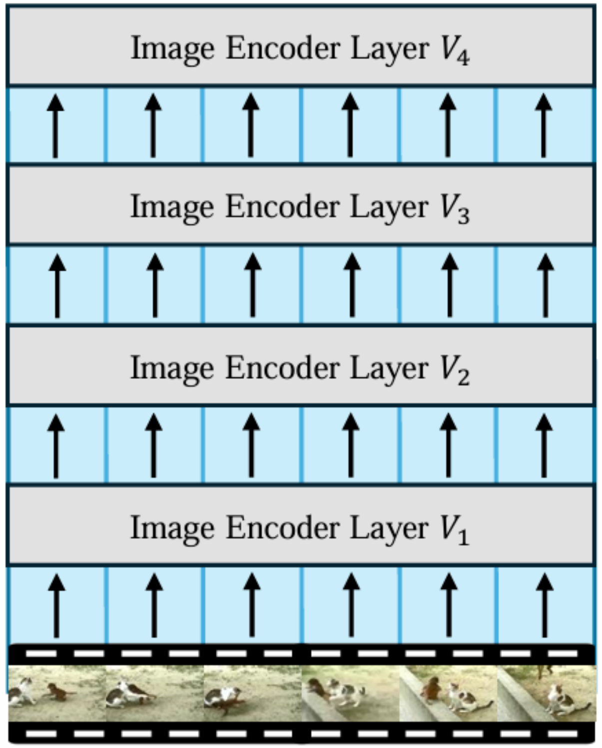
- The inherent differences between image and video modalities.
- Handling multiple sampled frames dramatically raises the number of patch tokens.

Challenges in trainable parameters.

- Current parameter-efficient fine-tuning methods incur high inference costs.

Challenges in model complexity.

- Current token compression methods overlook temporal redundancy in consecutive frames of a video.

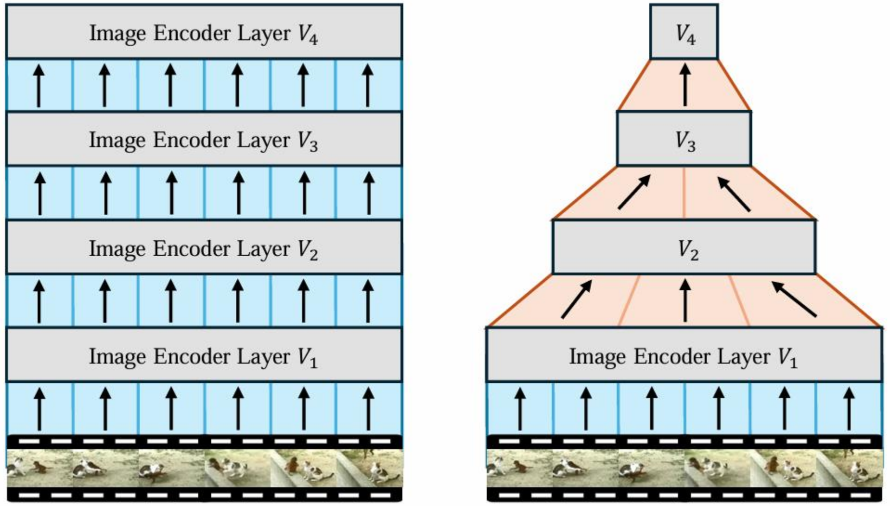


(b) Existing text-video retrieval methods

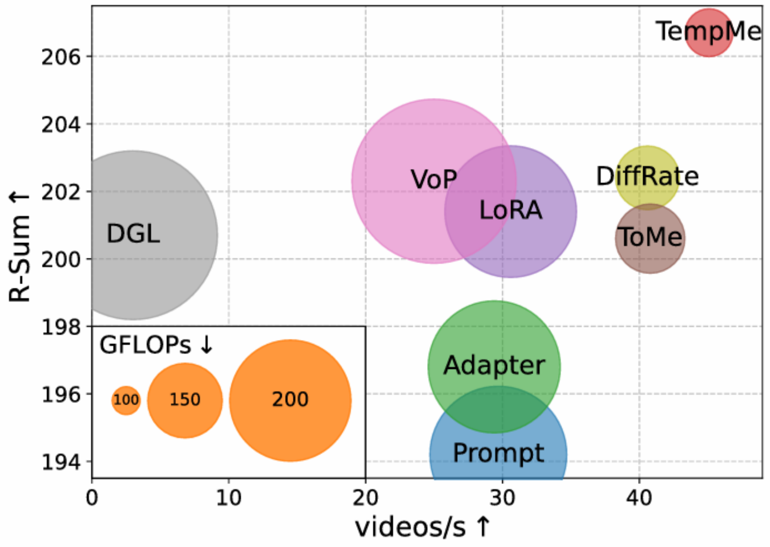
Background

We propose Temporal Token Merging (**TempMe**).

- A parameter-efficient and training-inference efficient TVR architecture that minimizes trainable parameters and model complexity.
- By gradually combining neighboring clips, we reduce spatio-temporal redundancy and enhance temporal modeling across different frames.
- Leading to improved efficiency and performance.



(b) Existing text-video retrieval methods. (c) Our TempMe.

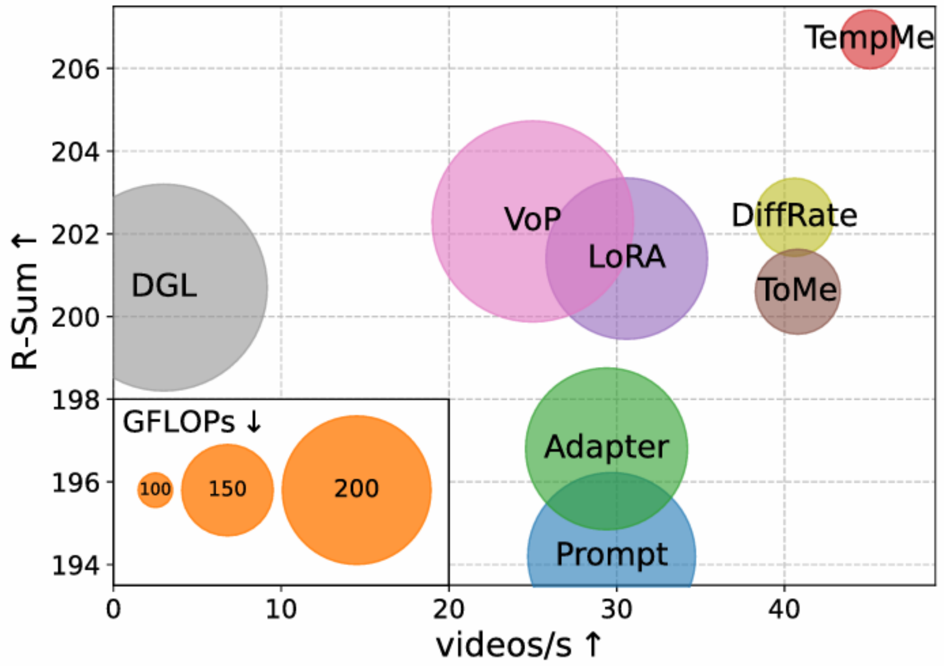


(d) Performance comparison.

Related Work

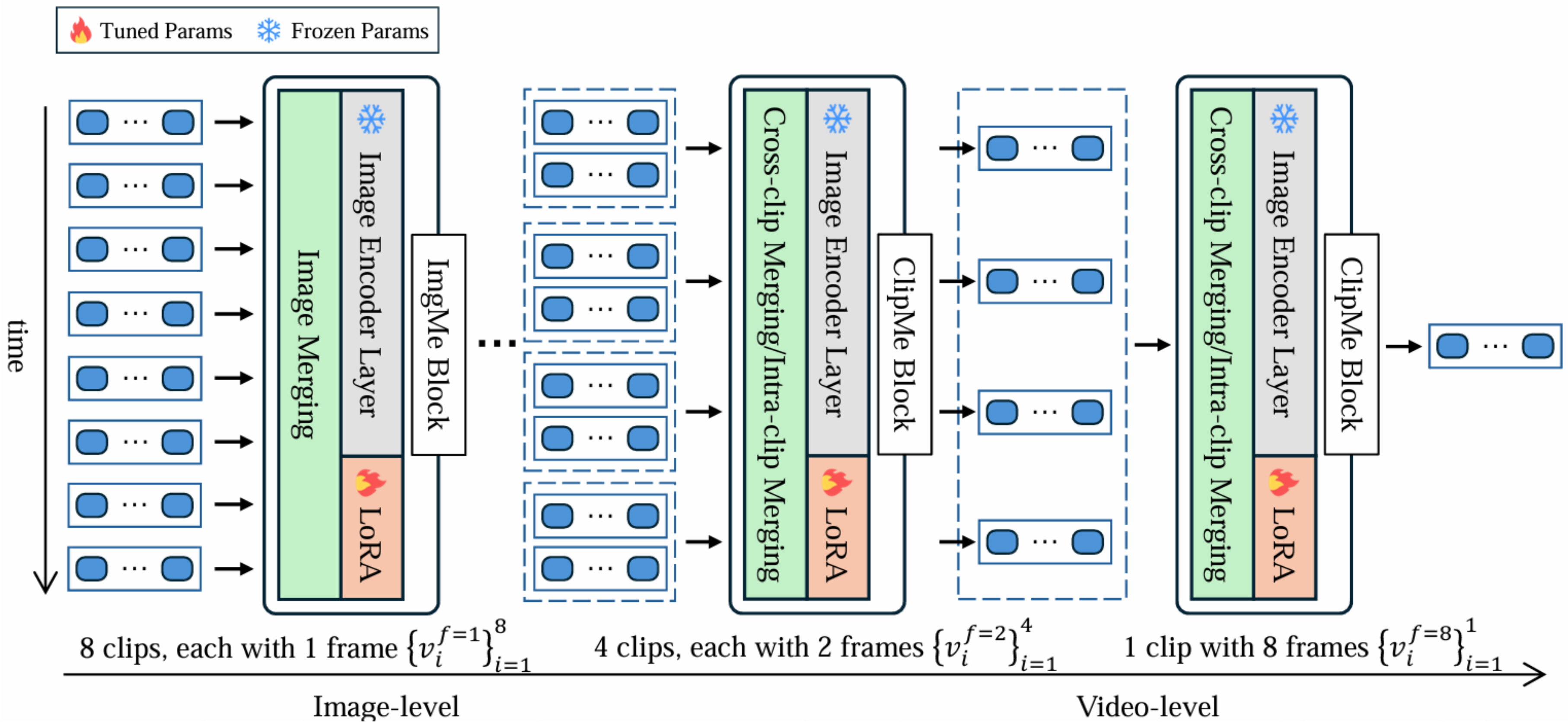
Compared with token compression methods.

- In CLIP-based text-video retrieval, each sampled frame is processed as an independent token set.
- Existing methods are limited to pruning tokens within a single token set for an image or video, without addressing token compression across multiple sets or incorporating temporal fine-tuning.
- Our TempMe fruitfully integrates parameter-efficient fine-tuning and token compression techniques, which minimizes spatio-temporal redundancy and enhance temporal modeling across frames.



(d) Performance comparison.

Meth

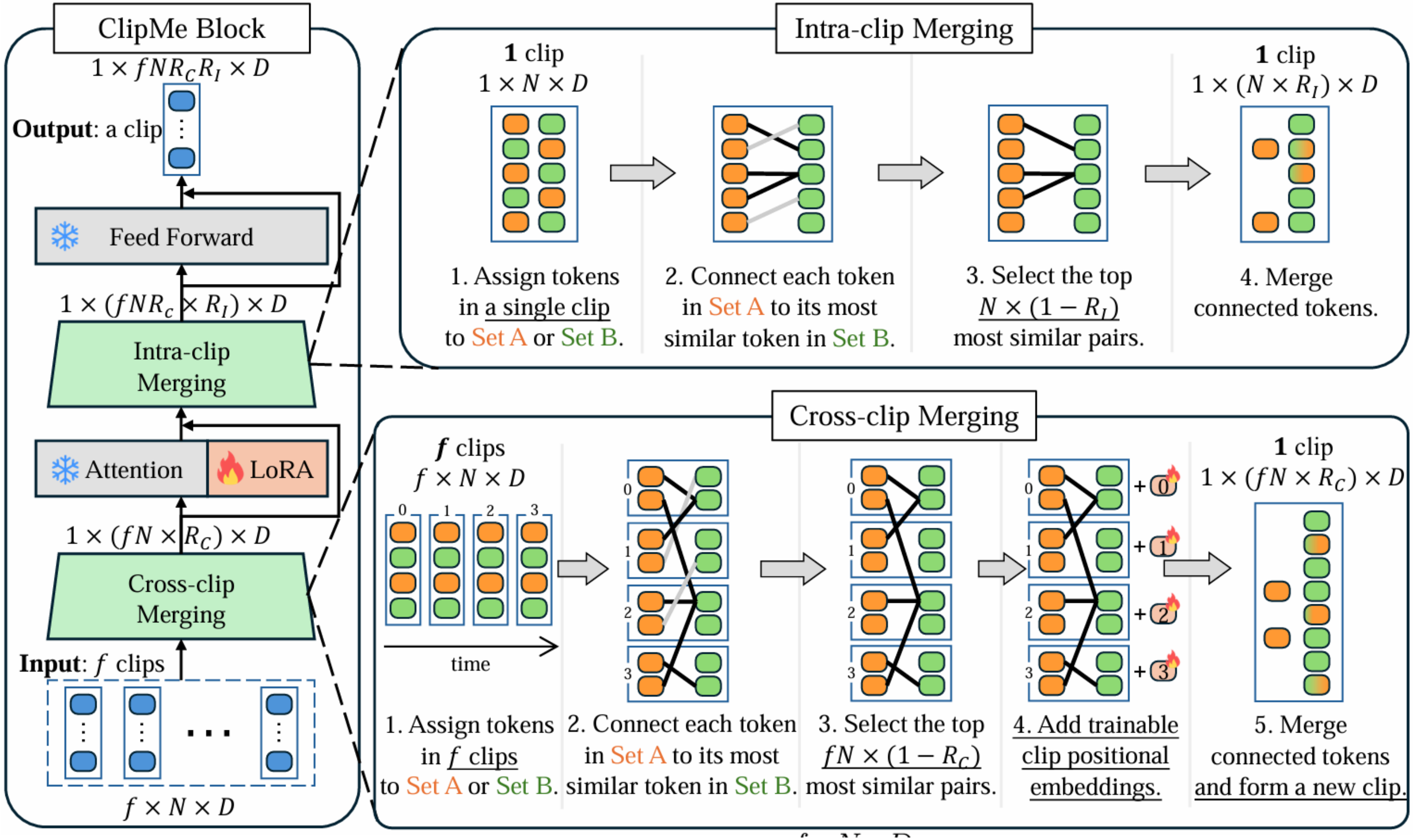


We freeze the pre-trained CLIP and merely train LoRA in both the image and text encoders.

We propose the Progressive Multi-Granularity framework.

- ImgMe Block independently encodes each single frame.
- ClipMe Block aggregates short-frame clips into extended-frame clips

Metho



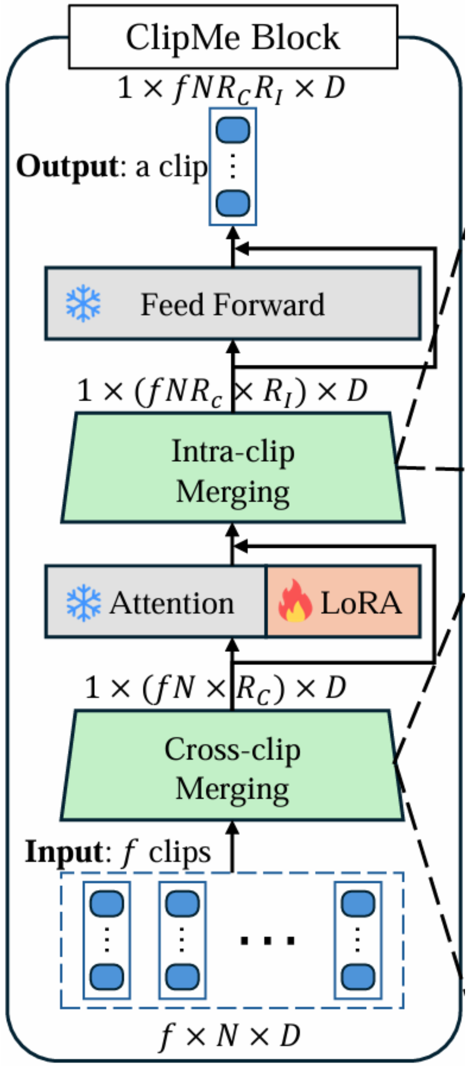
Instead of ImgMe Block which merges tokens in each single frame,

we propose ClipMe Block to process multi-frame clips.

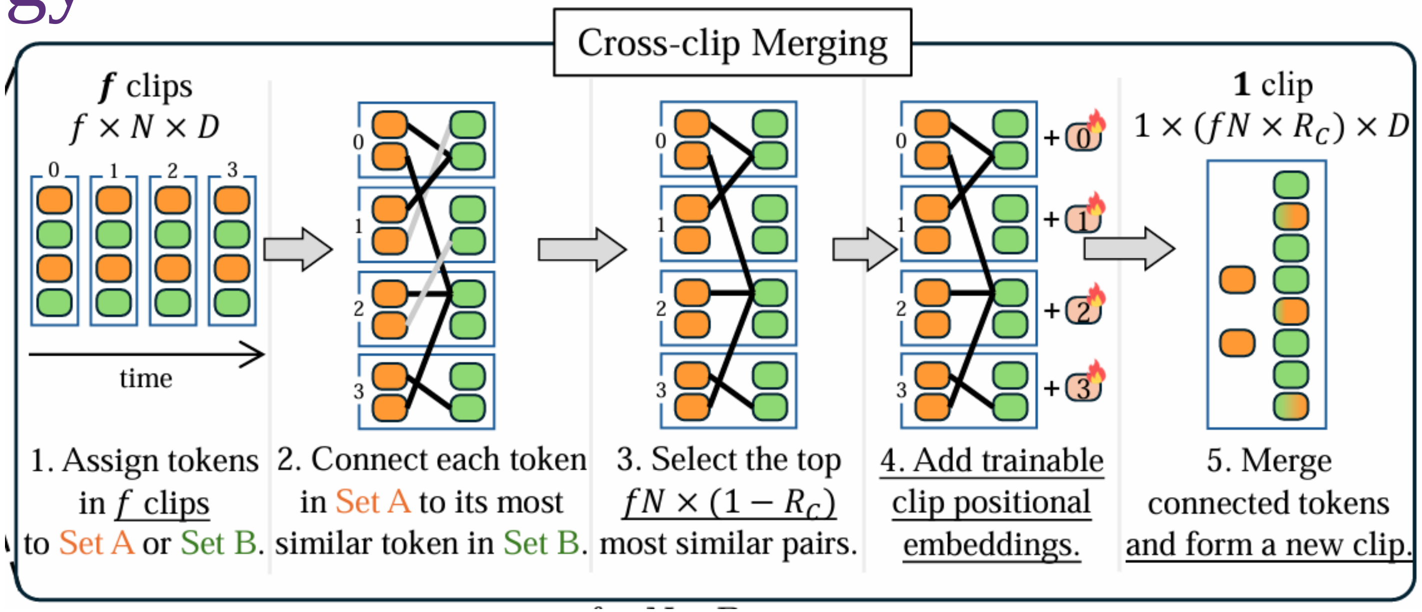
Methodology

ClipMe Block

- Cross-clip Merging:
  Adjacent clips are aggregated, which significantly reduces the number of temporal tokens and generates a new clip.
- Intra-clip Merging:
  The tokens within the newly formed clip are further compressed.



Methodology



Cross-clip Merging

- Considering the high temporal redundancy observed in videos, we select a large subset, $fN \times (1 - R_c)$, of the most similar token pairs for merging.
- Before merging, the trainable clip positional embeddings are added to facilitate temporal understanding.

Experiments

Complexity comparisons.

Backbone	CLIP-ViT-B/32						CLIP-ViT-B/16		
# Frames	12			64			12		
Method	GFLOPs	# Tokens	R@1/R-Sum	GFLOPs	# Tokens	R@1/R-Sum	GFLOPs	# Tokens	R@1/R-Sum
LoRA	53.0 (100%)	12 × 50 (100%)	43.7/193.0	276.7 (100%)	64 × 50 (100%)	38.7/191.5	211.3 (100%)	12 × 197 (100%)	47.3/201.4
DiffRate	36.8 (69%)	12 × 20 (40%)	41.5/189.9	190.1 (69%)	64 × 20 (40%)	38.0/188.7	138.5 (66%)	12 × 49 (25%)	47.3/202.4
ToMe	40.2 (76%)	12 × 26 (52%)	42.9/191.4	208.5 (75%)	64 × 26 (52%)	38.6/189.6	144.4 (68%)	12 × 77 (39%)	46.2/200.6
TempMe	**34.8** (**65**%)	**1** × **97** (**16**%)	**46.1/198.6**	**180.3** (**65**%)	**1** × **500** (**16**%)	**44.9/205.6**	**121.4** (**57**%)	**1** × **127** (**5**%)	**49.0/206.7**

Compared to ToMe,

our TempMe reduces tokens by more than **30**% for both CLIP-ViT-B/32 and CLIP-ViT-B/16, effectively decreasing model complexity while significantly surpassing accuracy.

With a 12-frame length and CLIP-ViT-B/16 backbone on MSRVTT,

TempMe outputs only **5**% of the input tokens, reaches **57**% GFLOPs, and achieves a **5.3**% R-Sum gain.

Experiments

Comparisons in the text-to-video task on MSRVTT.

Methods		# Params (M)	GFLOPs	R@1↑	R@5↑	R@10↑	R-sum↑	MnR↓
CLIP-ViT-B/32								
Full Fine-tuning	CLIP4Clip	123.54	53.0	43.1	70.4	80.8	194.3	16.2
Parameter-Efficient	Prompt	0.08	58.2	40.4	66.3	77.3	184.0	16.7
	Adapter	0.26	53.1	41.9	69.9	78.7	190.2	14.9
	LoRA	0.49	53.0	43.7	68.9	80.4	193.0	16.0
	PLEVU	6.35	-	36.7	64.6	76.8	178.1	-
	VoP^{F+C}	14.10	58.0	44.6	69.9	80.3	194.8	16.3
	DGL^{L}	0.83	67.4	44.7	70.5	79.2	194.4	16.2
	DGL^{T}	9.57	>67.4	45.8	69.3	79.4	194.5	16.3
Parameter-Efficient & Inference-Efficient	EVIT	0.49	37.2	41.4	69.0	78.1	188.5	16.8
	DiffRate	0.49	36.8	41.5	68.6	79.8	189.9	16.3
	STA	0.49	35.7	42.6	69.5	78.8	190.9	17.0
	ToMe	0.49	40.2	42.9	68.3	80.2	191.4	16.2
	TESTA	0.59	40.6	43.7	69.0	79.4	192.1	16.8
	TempMe	0.50	**34.8**	**46.1**	**71.8**	**80.7**	**198.6**	**14.8**
CLIP-ViT-B/16								
Parameter-Efficient	MV-Adapter	3.6	>210	46.0	72.0	82.1	200.1	-
	RAP	1.06	>210	46.5	73.9	82.0	202.4	12.1
	VoP	14.10	246.2	47.7	72.4	82.2	202.3	12.0
	DGL^{L}	0.83	251.2	48.3	71.8	80.6	200.7	13.4
Param&Infer-Efficient	TempMe	0.50	**121.4**	**49.0**	**74.4**	**83.3**	**206.7**	**11.9**

Our TempMe achieves significant improvements over previous methods,

with a **3.8**% R-Sum increase using ViT-B/32 and

a **4.3**% R-Sum increase using ViT-B/16,

while maintaining minimal GFLOPs.

Experiments

Previous parameter-efficient TVR, VoP and DGL, compromise efficiency for performance.

Unlike VoP and DGL,

our TempMe achieves a **1.8**× speedup over VoP and a **13.7**× speedup over DGL, while reducing GFLOPs by **51**% and improving R-Sum by **4.4**%.

Computational overhead comparisons of the CLIP-ViT-B/16 backbone.

Methods	videos/s	GFLOPs	# Tokens	R@1/R-Sum↑
Prompt	29.7	216.8	2369	44.3/194.2
Adapter	29.4	211.7	2364	44.9/196.8
LoRA	30.6	211.3	2364	47.3/201.4
VoP^{F+C}	25.0	246.2	2368	47.7/202.3
DGL^{L}	3.3	251.2	2416	48.3/200.7
DiffRate	40.6	138.5	588	47.3/202.4
ToMe	40.8	144.4	924	46.2/200.6
TempMe	**45.1**	**121.4**	**127**	**49.0/206.7**

THANKS!