Special Paper
방송공학회논문지 제30권 제7호, 2025년 12월 (JBE Vol. 30, No. 7, December 2025)
https://doi.org/10.5909/JBE.2025.30.7.1135
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

State Space Model based Temporal Adaptive Token Compression for Efficient Multimodal Large Language Model

Semi Kwon[a)] and Je-Won Kang[a)‡]

Abstract

With the recent surge in video content on digital platforms, there has been active research into applying the inference capabilities of large multimodal models (LMMs) to video understanding. However, the conventional approaches process videos as static frame sequences, resulting in ineffective utilization of temporal relationships between frames, which limits efficiency and performance. To address this issue, this study proposes a dynamic token compression module based on a selective state space model. The proposed method dynamically prunes visually redundant or low-priority video representations. It transmits only informative visual tokens to the large language model, thereby reducing computational load and memory usage while preserving core information. This paper presents an experiment on moment retrieval, where temporal precision and visual-linguistic alignment are more crucial than in existing token-based approaches such as ToMe and TokenLearner applied to the backbone. The results show that the proposed method achieves 2.04, 8.19, and 2.66%, 9.11%, and 0.84%, 3.86% higher performance than ToMe and TokenLearner in terms of mIoU, R1@0.5, and R1@0.7, respectively. This paper is based in part on the dissertation of the author [16].

Keyword : Multimodal AI model, State Space Model, Fine-tuning, Token Compression

I. Introduction

The recent growth of multimodal visual content has accelerated the adoption of large multimodal models (LMMs) in the field of video understanding. Transformer-based LMMs have achieved significant performance gains in visual-language tasks such as video captioning and video question answering by effectively aligning visual and textual features [1]. However, due to the limited number of input tokens, LMMs treat video as a simple sequence of image frames, which poses limitations in terms of repetitive input of similar visual information and the inability to exploit temporal relations between frames effectively. These drawbacks limit the improvement of video understanding performance [2].

a) Dept. of Artificial Intelligence and Software, Dept. of Electronic and Electric Engineering, Ewha Womans University
‡ Corresponding Author : Je-Won Kang
E-mail: jewonk@ewha.ac.kr
Tel: +82-2-3277-2347
ORCID: https://orcid.org/0000-0002-1637-9479
※ This research was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2024 (Project Number: RS-2024-00439534)
· Manuscript August 11, 2025; Revised September 23, 2025; Accepted September 23, 2025.

Token compression methods were proposed to address these problems. Video-ChatGPT [3] compressed video tokens using spatio-temporal pooling, and LLaVA-MR [4] proposed a dynamic compression approach after temporal encoding to input the final representation into LMMs. While existing approaches have alleviated computational complexity, they still lack a unified framework that jointly optimizes frame selection based on temporal dynamic modeling. Later, token reduction methods were developed in transformer-based vision processing. ToMe [5] combined redundant tokens through a bipartite matching based on attribute similarity. TokenLearner [6], on the other hand, proposed an adaptive tokenization approach that compresses hundreds of input tokens into 8 to 16 semantically important representations via learnable queries. They focused on removing unnecessary spatial information in patch tokens of the vision transformer (ViT).

In this paper, we propose a state space model-based temporal adaptive compression (STAC) module that selects temporally salient video representations along the temporal axis, while maintaining compatibility with existing pre-trained LMMs. Since projecting many video representations to the LLM can lead to computational overheads, our approach improves performance in inference while reducing computation and memory usage. Fig. 1 illustrates the token compression schemes of ToMe, TokenLearner, and our proposed module. The STAC dynamically excludes temporally redundant representations and passes only meaningful visual tokens to the LMM by employing the state space model as Mamba [8]. We conducted experiments on the temporal moment retrieval task, which demands high temporal precision and robust visual-language alignment. We applied ToMe and TokenLearner on the baseline to evaluate and compare different token compression methods, since they focused on removing redundancies in spatial tokens. These methods enable comparative studies of spatial and temporal token compression. Our experimental results clearly show that the proposed method significantly outperforms both ToMe and TokenLearner.

II. Related Works

1. Large Multimodal Model and Video Temporal Understanding

LMMs allow comprehensive video understanding by

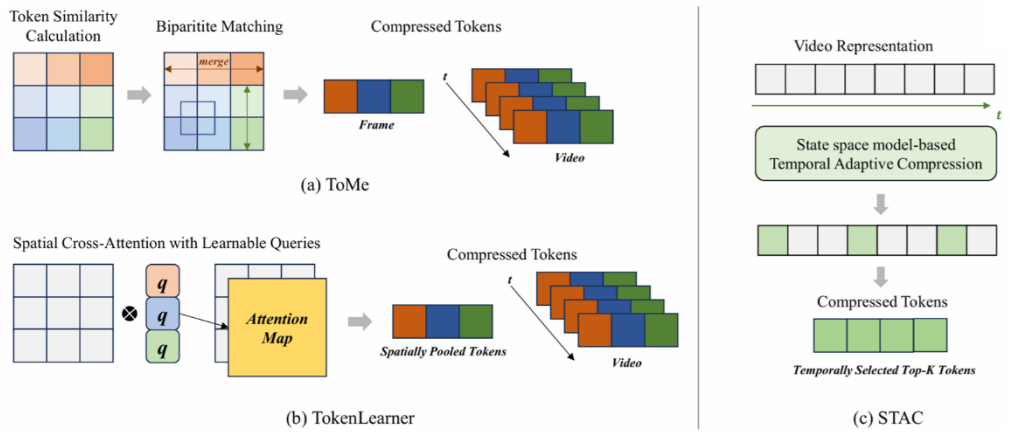


Fig. 1. Comparisons of token compression methods of (a) ToMe, (b) TokenLearner, and (c) our module, STAC. ToMe and TokenLearner focus on spatial token compression in the vision transformer. In contrast, STAC addresses video representation in Top-K temporal token selection to use the state space model for enhancing compatibility with pre-trained models.

combining visual, textual, and auditory features [11]. Earlier approaches introduced projection layers and Q-Former to map the output from visual encoders, incorporating additional parameters such as gated cross-attention and adapter layers, to the baseline models to process multimodal inputs [12, 13]. However, token limitations of the LLMs force the adoption of sparse sampling, which inherently overlook fine-grained temporal information and immediate visual details [2].

Several studies have proposed adaptive frame processing techniques to overcome these limitations. Video-ChatGPT [3] introduced the frame compression method that utilizes spatio-temporal pooling to aggregate information efficiently. LLaVA-MR [4] proposed an approach to capture detailed temporal representation through time encoding and select informative frames using a variance-based dynamic compression. While these approaches have partially alleviated the problem of limited token input, a lack of integrated processing remains that leverages the interdependence between frame selection and the temporal context of the video.

2. Token Compression for Video Representation

An efficient token reduction approach is essential in video processing due to the quadratic computational complexity of transformers. ToMe improves throughput by a factor of two without retraining by combining duplicate tokens using a bipartite matching algorithm based on attention key similarity. TokenLearner proposes an adaptive tokenization method that compresses hundreds of input tokens into 8–16 semantically salient representations using an attention mechanism.

In this study, the STAC module dynamically evaluates the saliency of visual tokens generated by the Q-Former of the LMM before projecting them to the backbone, by selectively compressing the core information. Through the long-term dependency modeling of Mamba, the STAC compresses visual tokens based on global temporal context, enabling efficient computation without sacrificing semantically critical visual details.

III. Methods

1. Model Architecture

We introduce a lightweight video temporal understanding framework, as shown in Fig. 2. The extracted vid-

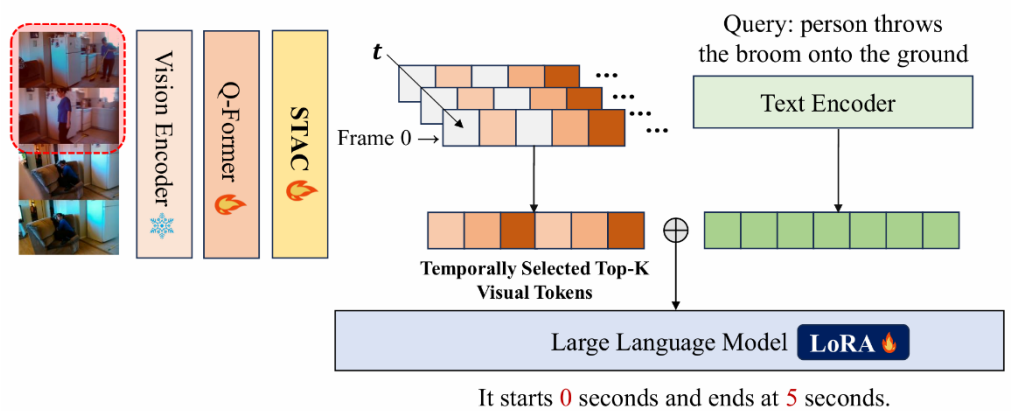


Fig. 2. We use a 4-bit quantized BLIP-2 [9] that encodes video and text queries. The videos are embedded via a frozen vision encoder and processed through Q-Former to produce video representations. STAC reduces the 32 dimensions of the Q-Former outputs to 16 temporally salient tokens. After projection and concatenation with text embeddings, the model processes these through a LoRA-adapted LLM [13] for end-to-end fine-tuning.

eo representation from Q-Former is processed along the temporal axis using a compression module. The STAC effectively reduces the sequence length of vision embeddings by extracting only the salient tokens. This approach reduces the dimensionality of the Q-Former output and significantly shortens the LLM [13] input sequence length, thereby mitigating the input bottleneck while improving computational efficiency and reducing inference time.

First, the vision encoder extracts uniformly sampled frames, and then reconstructed visual pixel values are processed to generate a frame-by-frame embedding. It processes $[B \times N, C, H, W]$, where the dimensions represent batch size, number of frames, channels, height, and width. The output frame embeddings of the vision encoder contain the patch tokens for each frame and finally consist the form $F \in R^{B \times N \times P \times D}$, where P is the number of patch tokens per frame, and D is the embedding dimension. They are flattened to a single sequence and fed into Q-Former. Q-Former utilizes learnable query tokens to extract high-level features. Our proposed Mamba-based token compression processes the video representation of the Q-Former, which selects the most salient Top-K tokens along the temporal axis of the form $[B \times N \times K \times D]$.

The text encoder processes the target query embedding form of $T \in R^{B \times L \times D}$, where L denotes the sequence length. The visual tokens selected by the STAC are mapped to the same dimensional space as the text embeddings through a projection layer, after which the two modalities construct an integrated multimodal representation. Finally, the LLM generates the temporal localization output through the integrated representation in the following format: “It starts at {start time: X} and ends at {end time: Y} seconds.”

2. State space model-based Temporal Adaptive Compression

The STAC selects temporally salient video representation tokens from Q-Former outputs for efficient fine-tuning while preserving the knowledge of pre-trained LMM models. Projecting the large number of Q-Former output tokens into the LLM input causes a computational bottleneck. Our approach addresses it by selectively retaining only the most temporally salient tokens. STAC prunes less informative tokens by leveraging attention and inter-frame relational analysis and decreases the amount of computation after projection without compromising the saliency of the video tokens.

Fig. 3 presents the detailed architecture of the STAC module. The input of STAC is $[B \times N \times Q \times D]$, where Q indicates the number of Q-Former output tokens, and D corresponds to the feature dimensionality of each token. The original representation is reshaped into a sequential format $[B \times Q,\ N,\ D]$ through a view operation. We add a positional encoding, normalize the value, and input into

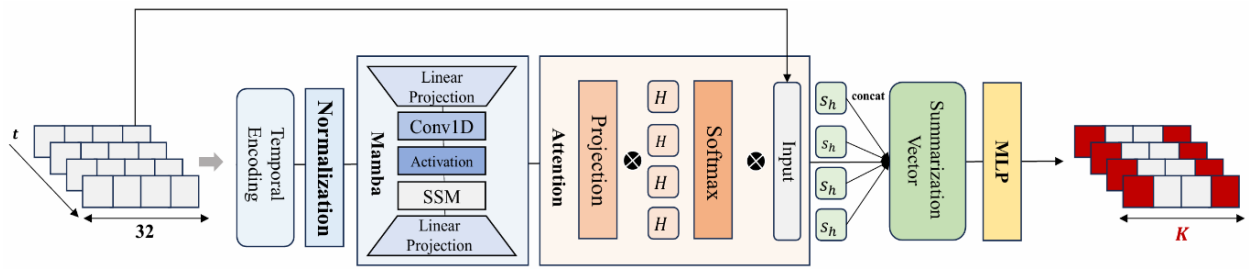


Fig. 3. The Q-Former outputs receive temporal position encoding before Mamba processing to model sequence dynamics. The video tokens undergo saliency assessment through an attention layer with learnable queries. Based on the computed saliency rankings, the module retains only the top-K most relevant tokens for subsequent stages.

the Mamba block. Mamba employs an input-conditioned transition matrix to encode sequential patterns across N time steps into a compressed state representation, preserving salient information while eliminating redundant components [8].

After generating the key tensor $[B \times Q,\ N,\ D]$ via projection, we calculate the attention weights $[B \times Q,\ H,\ D]$ using four learnable queries $[H,\ D]$. Applying softmax over the temporal axis of N, each attention head generates a normalized weight distribution that highlights salient temporal localization in the sequence. Each head produces a summary vector that captures temporal information, resulting in head-specific features $[B \times Q,\ H,\ D]$. These are flattened $[B \times Q,\ H \times D]$ MLP to output a scalar score. Within each batch, we apply Top-K filtering to isolate the highest-scoring query localization. These positions are arranged in chronological order to ensure sequence integrity and facilitate feature extraction from the original embedding space. This process results in an output embedding with dimensions $[B, N,\ K, D]$

3. Training and Inference

We employ the STAC to extract the key visual tokens that compose the video representation, reducing computation by quantizing the pre-trained model to 4-bit and incorporating low-rank matrices into some LLM layers with LoRA for efficient training. LoRA preserves the weight W_0 and only trains the low-rank update weights $\Delta W = BA$. LoRA achieves video-to-text alignment by fine-tuning visual and textual tokens together through low-rank matrix decomposition, which preserves the pretrained weights while generating fine-tuned weight $W_0 + \Delta W$ applied during inference for efficient processing [10].

We define the following loss function for fine-tuning.

$$L(x, y) = \ -\sum_{i=1}^{l-1} log_{p_\theta}\,(y_{i+1}|x, y_{1:i}),$$

where we use the model maximum likelihood loss minimization over input x and target y pairs of length l, then use greedy decoding during inference to generate temporal descriptions from which predicted start and end times are parsed and compared against ground truth for performance evaluation.

Ⅳ. Experiment

1. Experimental Setup

Datasets - we assess the performance of our module with Charades-STA [14], which serves as the primary evaluation for temporal localization tasks. It comprises 16,128 annotated samples containing segments with an average length of 8.1 seconds and text queries consisting of 7.22 words on average, extracted from videos with a mean duration of 30.6 seconds. This distribution includes 12,408 training samples and 3,820 test samples.

Metric - we employ conventional assessment for video temporal grounding of Recall@K and Intersection over Union (IoU) metrics. The mean IoU(mIoU) computes the average overlap ratio between predicted and ground truth across all samples. Recall@K quantifies the fraction of Top-K predictions that surpass a specified IoU threshold. We report R1@0.5 and R1@0.7 as accuracy measures for IoU thresholds of 0.5 and 0.7, respectively.

Implementation Details - we fine-tune 1.22% of the total parameters. We employ the AdamW optimization combined with WarmupCosineLR scheduling, configuring the learning rate at 1e-5 and weight decay at 0.01. The learning rate schedule incorporates a linear warmup phase that gradually increases from zero to peak value during the initial 10% of training iterations, followed by the cosine decay over the remaining 90%. We conduct experiments on a sin-

gle A6000-48GiB GPU, processing 16 frames per video sequence and implementing early stopping across 20 training epochs. The training process identified epoch 11 as the optimal stopping point due to overfitting in later epochs, with the entire training approximately 19 hours. We hold the same experimental conditions in the tested methods for fair comparisons.

2. Performance Evaluation

2.1 Performance Comparison with Existing Approaches

In this study, we evaluate the performance and efficiency of our proposed STAC method within an LMM [9] architecture. We compare it against conventional token compression approaches: ToMe [5] and TokenLearner [6]. While ToMe merges similar vision transformer tokens and TokenLearner employs learnable queries for compression, STAC takes a different approach, which selects the Top-K most temporally salient tokens from the final concatenated video representation before feeding them into the LLM [13].

Table 1. Performance Comparison with Existing Approaches

Method	mIoU	R1@0.5	R1@0.7
TokenLearner	22.1	25.2	13.2
ToMe	28.2	31.7	16.2
Ours (STAC)	30.2	34.3	17.1

The STAC method demonstrated superior performance. It achieved an mIoU of 30.2, R1@0.5 of 34.3, and R1@0.7 of 17.1. In contrast, conventional token compression methods vary in degrees of performance degradation. TokenLearner exhibited the most significant performance degradation, with its mIoU of 22.1%, R1@0.5 of 25.2, and R1@0.7 of 13.2%. ToMe, while outperforming TokenLearner, still showed a degraded performance with mIoU with 28.2, R1@0.5 of 31.7, and R1@0.7 of 16.2.

We evaluated the computational efficiency of the model using FLOPs, MACs, and number of parameters, as summarized in Table 2. FLOPs and MACs directly represent the computational complexity of the model and were measured using the Calflops package. When TokenLearner was applied, TFLOPs increased slightly to 15.57 and TMACs to 7.78, indicating that the module imposes computational overhead. ToMe showed the highest values with 16.59 TFLOPs and 8.30 TMACs, suggesting that the token merging process caused higher-than-expected computational costs. On the other hand, the STAC achieved the lowest computational complexity with 12.51 TFLOPs and 6.25 TMACs. When comparing the number of model parameters, TokenLearner had the highest at 151.65 million, followed by STAC with 149.48 million. ToMe maintained the same 143.67 million parameters. While parameters of STAC are slightly higher than ToMe, it is lower than TokenLearner. It suggests an appropriate level of overhead for balancing efficiency and performance improvements.

Table 2. Computational Complexity and Efficiency Comparison

Method	FLOPs (TFLOPs)	MACs (TMACs)	Parameters (M)
TokenLearner	15.57	7.78	151.66
ToMe	16.59	3.30	143.67
Ours (STAC)	12.51	6.25	149.48

In addition, we report inference time. TokenLearner used 0.37 seconds, and ToMe used 0.35 seconds, showing slight improvements. Due to a reduction in attention computations, it results from a decrease in the number of tokens. STAC recorded the fastest inference time of 0.30 seconds all methods. It demonstrates that enhancing computational efficiency leads to improved inference speed and reduced resource consumption.

2.2 Performance Analysis

To determine the optimal number of temporal tokens for our STAC framework, we conduct an experiment examining the impact of Top-K selection on model performance. We evaluate K = {4, 8, 16}, representing different degrees of temporal compression.

Table 3. Top-K Selection Analysis

Top-K	4	8	16
mIoU	31.09	27.58	31.28
R1@0.5	33.66	30.22	34.21
R1@0.7	18.25	15.19	18.33

The experimental results in Table 3 show the results across different K values. K=4 achieves an mIoU of 31.09, while K=8 shows a significant performance drop to 27.58. K=16 recovers to achieve the highest performance at 31.28. At K=4, the model demonstrates stable performance through complete compression, selecting only the most essential temporal information. This aggressive compression eliminates redundancy while maintaining core temporal patterns. The intermediate value K=8 yields the poorest performance across all metrics, representing a suboptimal region where the model fails to make an appropriate balance between temporal compression efficiency and comprehensive temporal representation. K=8 provides insufficient tokens for capturing the full temporal diversity needed for complex moment localization, while simultaneously offering too many tokens to achieve the focused compression benefits seen at K=4. At K=16, the model achieves the best overall performance, particularly excelling in precise matching tasks (R1@0.5: 34.21). This configuration provides sufficient temporal diversity and coverage, enabling comprehensive preservation of temporal relationships across different time scales. These findings indicate that effective temporal compression operates optimally at two distinct regimes: either complete compression (K=4) for maximum efficiency, or sufficient expansion (K=16) to guarantee comprehensive temporal coverage. Based on these results, we adopt 16 as our final configuration, as it achieves the highest performance while providing sufficient temporal expressiveness for complex video understanding tasks.

Ⅴ. Conclusion

Recent LMMs struggle with video temporal understanding due to redundant visual information and limitations in efficient temporal reasoning. We proposed State Space Model-based Temporal Adaptive Compression (STAC), a novel module that leverages Mamba-based state-space models to dynamically select temporally salient video representations while excluding redundant tokens, addressing these limitations. Our experiments on moment retrieval tasks demonstrate the superior performance of STAC over existing compression methods such as ToMe and TokenLearner, achieving notable improvements in mIoU, R1@0.5, and R1@0.7 metrics. In addition, STAC significantly reduces computational complexity with lower TFLOPs, faster inference times, and reduced memory usage. STAC represents a significant advancement in efficient video understanding, providing a computationally optimized solution that maintains compatibility with pre-trained models while enabling more effective video content analysis.

Reference

[1] Cao, Z., Zhang, B., Du, H., Yu, X., Li, X., & Wang, S. (2025, February). Flashvtg: Feature layering and adaptive score handling network for video temporal grounding. In 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (pp. 9226-9236). IEEE.
doi: https://doi.org/10.1109/wacv61041.2025.00894

[2] Meinardus, B., Batra, A., Rohrbach, A., & Rohrbach, M. (2024). The surprising effectiveness of multimodal large language models for video moment retrieval. arXiv preprint arXiv:2406.18113.
doi: https://doi.org/10.48550/arXiv.2406.18113

[3] Lu, W., Li, J., Yu, A., Chang, M. C., Ji, S., & Xia, M. (2024). LLaVA-MR: Large Language-and-Vision Assistant for Video Moment Retrieval. arXiv preprint arXiv:2411.14505.
doi: https://doi.org/10.32388/vlxb6m

[4] Maaz, M., Rasheed, H., Khan, S., & Khan, F. S. (2023). Video-chatgpt: Towards detailed video understanding via large vision and language models. arXiv preprint arXiv:2306.05424.
doi: https://doi.org/10.18653/v1/2024.acl-long.679

[5] Bolya, D., Fu, C. Y., Dai, X., Zhang, P., Feichtenhofer, C., & Hoffman, J. (2022). Token merging: Your Vit But Faster. arXiv preprint arXiv:2210.09461.
doi: https://doi.org/10.48550/arXiv.2210.09461

[6] Ryoo, M. S., Piergiovanni, A. J., Arnab, A., Dehghani, M., & Angelova, A. (2021). Tokenlearner: What can 8 learned tokens do for images and videos?. arXiv preprint arXiv:2106.11297.
doi: https://doi.org/10.48550/arXiv.2106.11297

[7] Jiang, J., Li, X., Liu, Z., Li, M., Chen, G., Li, Z., ... & Byeon, W. (2025). Token-efficient long video understanding for multimodal llms. arXiv preprint arXiv:2503.04130.
doi: https://doi.org/10.48550/arXiv.2503.04130

[8] Gu, A., & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752.
doi: https://doi.org/10.48550/arXiv.2503.04130

[9] Li, J., Li, D., Savarese, S., & Hoi, S. (2023, July). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In International conference on machine learning (pp. 19730-19742).
doi: https://doi.org/10.48550/arXiv.2301.12597

[10] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms, 2023. URL https://arxiv.org/abs/2305.14314, 2.
doi: https://doi.org/10.48550/arXiv.2305.14314

[11] Lin, B., Ye, Y., Zhu, B., Cui, J., Ning, M., Jin, P., & Yuan, L. (2023). Video-llava: Learning united visual representation by alignment before projection. arXiv preprint arXiv:2311.10122.
doi: https://doi.org/10.18653/v1/2024.emnlp-main.342

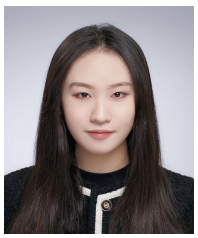
[12] Jin, Y., Li, J., Liu, Y., Gu, T., Wu, K., Jiang, Z., ... & Ma, L. (2024). Efficient multimodal large language models: A survey. arXiv preprint arXiv:2405.10739.
doi: https://doi.org/10.48550/arXiv.2405.10739

[13] Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., ... & Wei, J. (2024). Scaling instruction-finetuned language models. Journal of Machine Learning Research, 25(70), 1-53.
doi: https://doi.org/10.48550/arXiv.2210.11416

[14] Gao, J., Sun, C., Yang, Z., & Nevatia, R. (2017). Tall: Temporal activity localization via language query. In Proceedings of the IEEE international conference on computer vision (pp. 5267-5275).
doi: https://doi.org/10.1109/iccv.2017.563

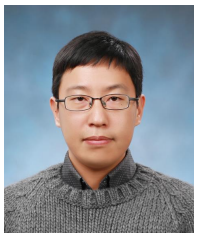
[15] Semi Kwon, Temporal Contextual Fine-tuning of Multimodal Large Language Models using Dynamic Adaptive Video Representation Compression for Highlight Moment Detection (Master Degree dissertation, Ewha Womans University, 2025.06). https://dspace.ewha.ac.kr/handle/2015.oak/272630

Introduction Authors



Semi Kwon

- Present : Dept. of Artificial Intelligence and Software, Dept. of Electronic and Electric Engineering, Ewha Womans University
- ORCID : https://orcid.org/0009-0004-7472-0438
- Research interests : Multimodal AI, Video Temporal Localization



Je-Won Kang

- 2008.8 ~ 2012.7 : Ph.D, University of Southern California
- 2012.8 ~ 2014.2 : Senior engineer, Qualcomm Inc.
- 2014.3 ~ Present : Professor, Ewha Womans University
- ORCID : https://orcid.org/0000-0002-1637-9479
- Research interests : Multimodal AI model, Video coding and processing