Towards Efficient Video LLMs via Visual Token Compression

2026년도 하계 세미나

Sogang University
Vision & Display Systems Lab, Dept. of Electronic Engineering

Presented By
남준식

Outline

- Background
  - Video large language model
  - Video token compression

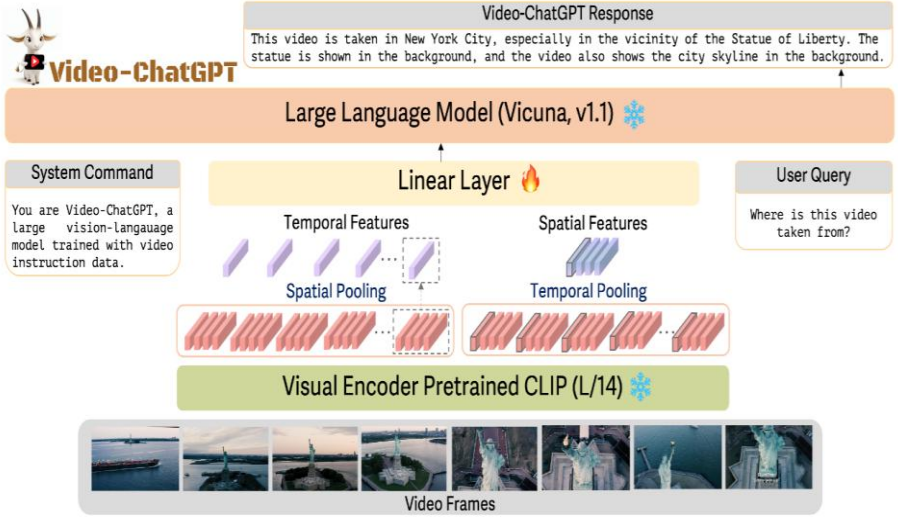
- 논문 선정의 이유

- Unified Spatiotemporal Token Compression for Video-LLMs at Ultra-Low Retention
  - CVPR 2026

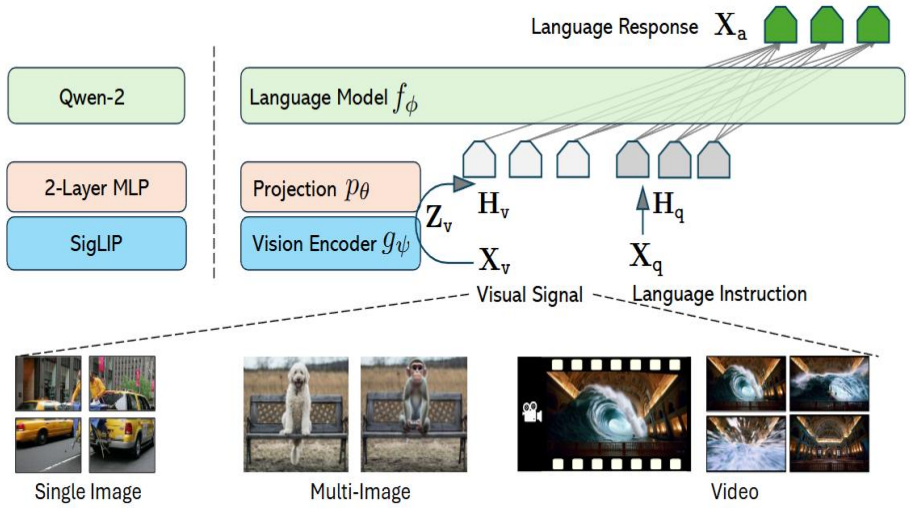
- KiToke: Kernel-based Interval-aware Token Compression for Video Large Language Models
  - Arxiv 2026

Background

- Video Large Language model(Video LLM)
  - Video LLM은 VLM과 동일하게 encoding → projectio → LLM 구조 가짐
    - 입력 vision data를 시간 축을 갖는 비디오 표현으로 확장한 모델
    - 비디오를 여러 frame으로 sampling 한 뒤, 각 frame을 vision encoder로 독립적으로 처리
      - 각 프레임이 독립적으로 encoding 되기에, 프레임 간의 관계를 모델링 하지 못함
    - 대표적으로 video-chatgpt, llava-onevision, llava-video 모델이 존재



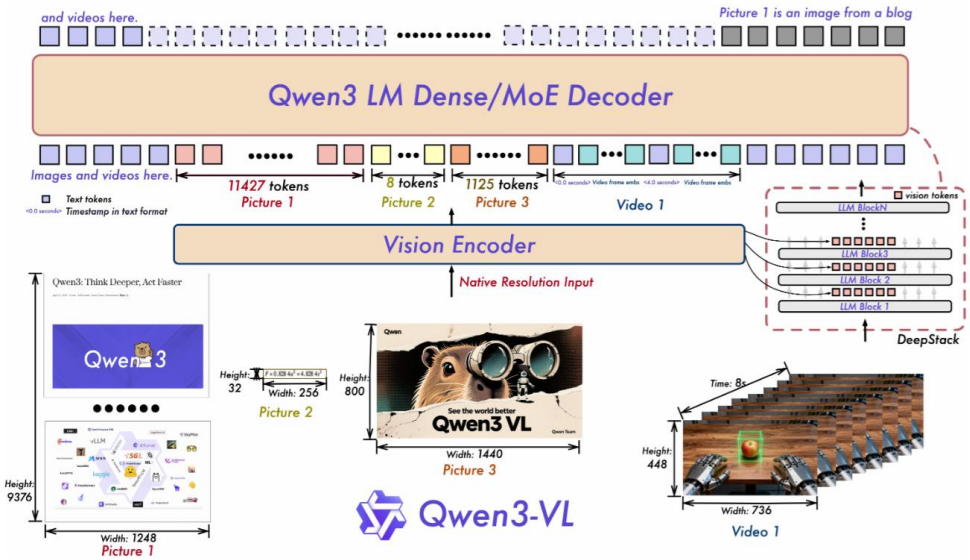
< Video-ChatGPT Framework >



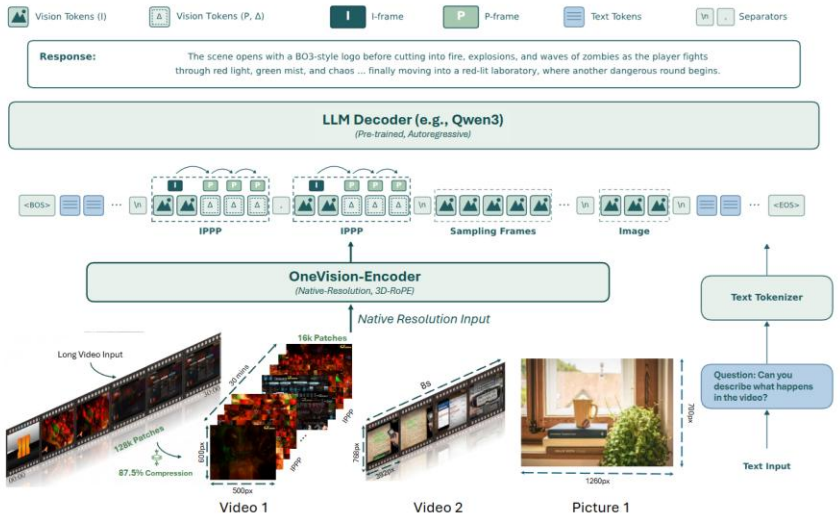
< LLava-onevision Framework >

Background

- Video Large Language model(Video LLM)
  - Video LLM with Spatiotemporal encoder
    - Vision encoder 에서 frame간의 관계를 직접적으로 모델링 하는 Video LLM
      - 3D Conv를 통하여 인접 프레임의 spatial patch를 spatiotemporal token으로 변환
      - 3D RoPE을 사용하여 각 vision token에 시간 및 공간 위치를 함께 부여
    - 대표적으로 Qwen2.5-VL, Qwen3-VL, Llava-Onevision-2 모델이 존재



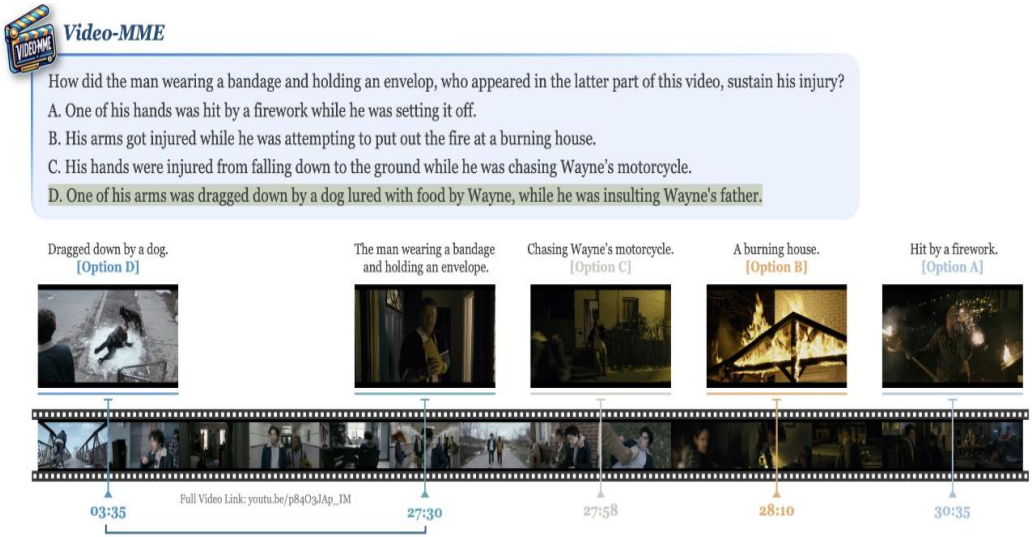
< Qwen3-VL Framework >



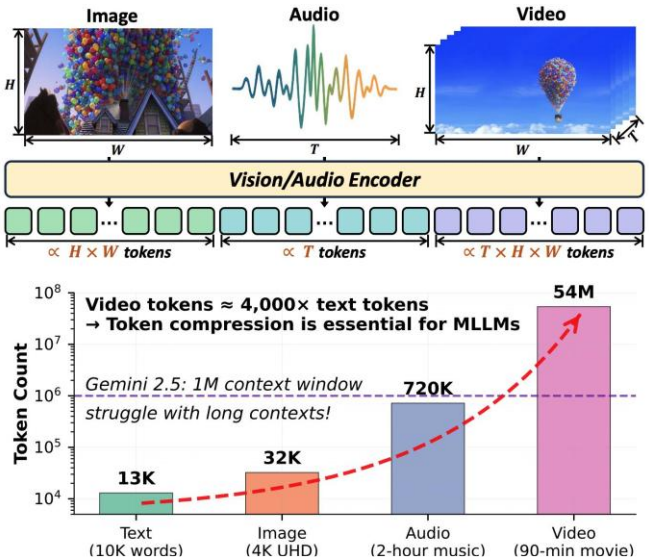
< LLava-onevision-2 Framework >

Background

- Video Large Language model(Video LLM)
  - Benchmark for Video LLMs
    - Video LLM의 평가를 위해서 입력 비디오에 대해서 Video QA를 통해서 성능 평가
    - 대표적으로 Video-MME, MVbench, MLVU, LongvideoBench 가 존재
  - Computational Limits of Video LLMs
    - LLM의 제한된 context length로 인해 입력으로 들어가는 token budget이 제한
    - 많은 visual token은 LLM의 prefill 연산량과 KV-cache memory를 증가



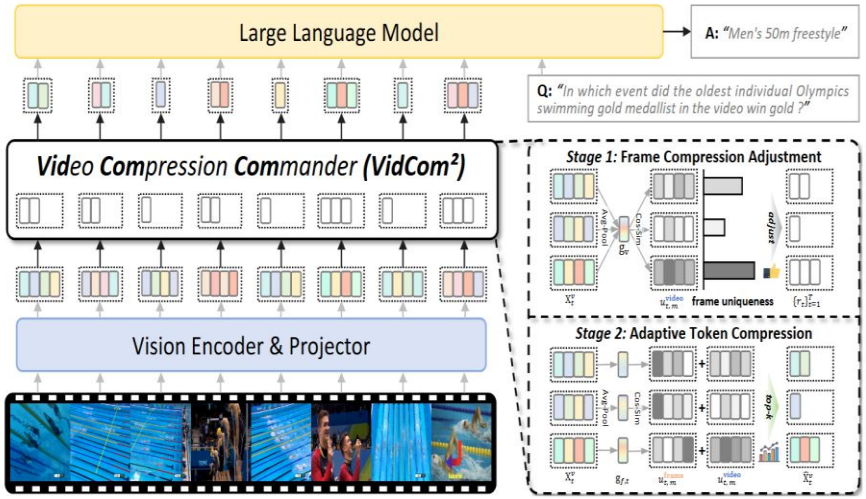
< Example case of Video-MME benchmark >



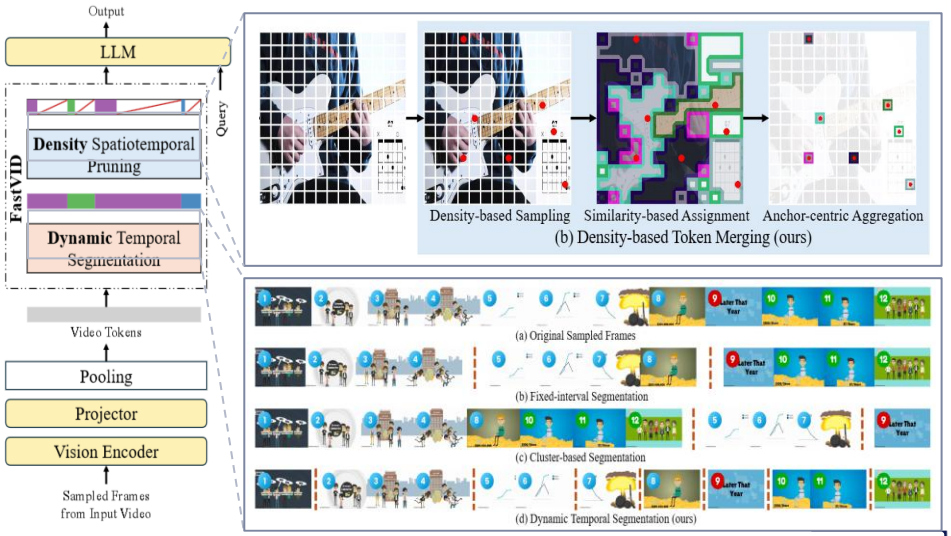
< Computational limits of Video LLMs >

Background

- Video token compression
  - Pre-LLM token compression
    - Token compression을 LLM 입력 이전에 적용하는 방법
      - Vision Encoder가 생성한 visual token 중 중요한 token만을 선택하여 LLM에 전달
        - ✓ 입력 데이터의 특성을 반영하여 중요한 token의 특징을 정의
      - LLM architecture를 수정하지 않는 plug-and-play 적용이 가능
    - 대표적으로 VidCom²[ACL 25], Fastvid[NeurIPS 25], KiToke[ICML 26] 가 존재



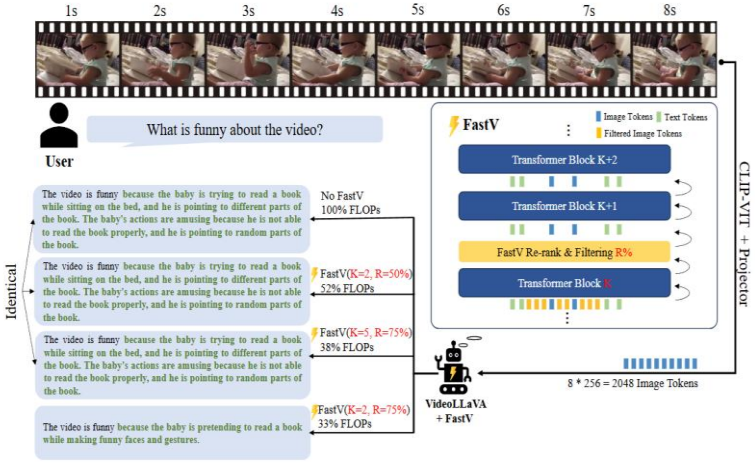
< VidCom² [ACL 25] Framework>



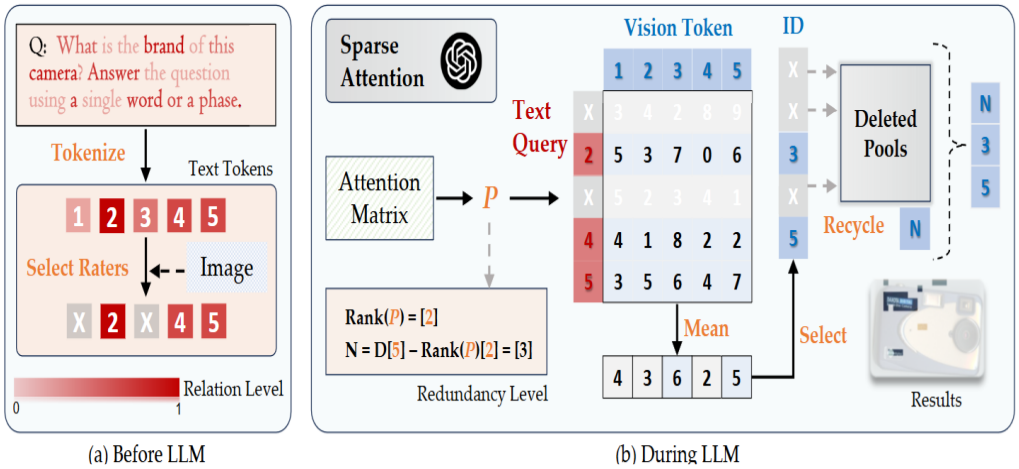
< Fastvid[NeurIPS 25] Framework>

Background

- Video token compression
  - Inner-LLM token compression
    - Token compression을 LLM 입력 이후에 적용하는 방법
      - LLM의 attention map에서의 특성을 이용하여 불필요한 token을 줄이는 방법
      - Text–visual interaction을 활용하므로 query-aware한 token selection이 가능
        - ✓ 초기 LLM layer에서는 모든 visual token을 처리해야 하는 한계가 존재
          - 유의미한 attention map의 경우 후반부 LLM layer에서 형성 되기 때문
    - 대표적으로 FastV[ECCV 24], SparseVLM[ICML 25] 가 있음



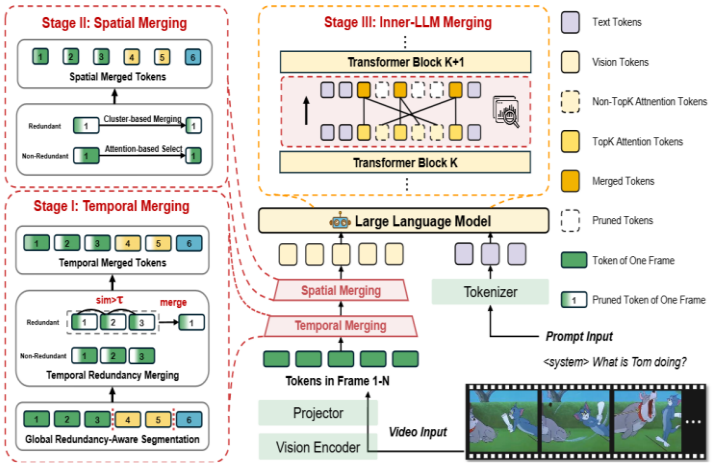
< FastV [ECCV 24] Framework >



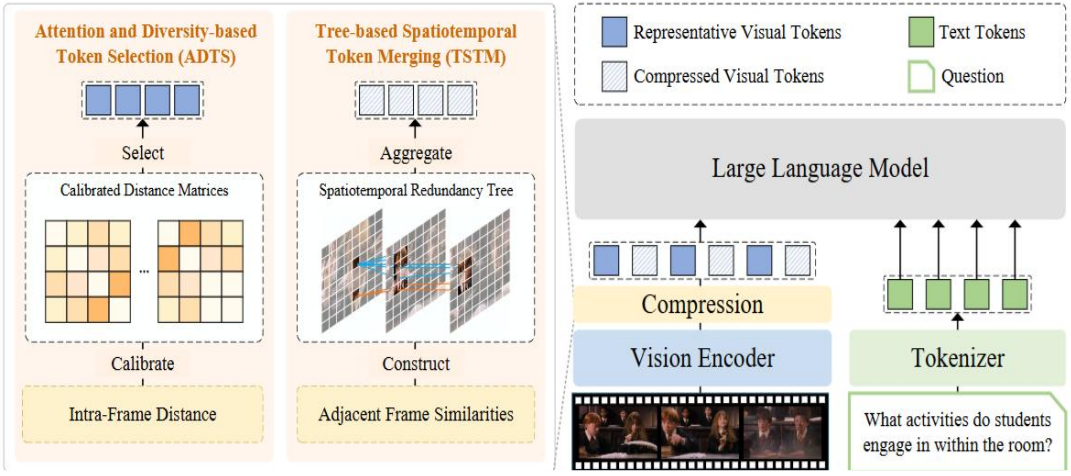
< Sparse VLM[ICML 25] Framework >

Background

- Video token compression
  - Hybrid token compression
    - Pre-LLM 및 Inner-LLM 방법을 함께 적용하는 방법
    - 입력 비디오의 redundancy를 먼저 제거한 뒤, query relevance를 기반으로 추가 압축
      - Visual redundancy와 query relevance를 모두 활용하기에 높은 압축률에서도 중요한 token을 보존하는 데 유리
      - 두 번의 compression 단계에서 비롯된 computational overhead가 존재
    - 대표적으로 HoliTom[NeurIPS 25], FlashVid[ICLR 26] 가 있음



< HoliTom[NeurIPS 25] Framework >



< FlashVid[ICLR 26] Framework >

Background

- 논문 선정의 이유
  - 극단적인 token compression 에 있어서 Token selection을 위 고려사항 확인

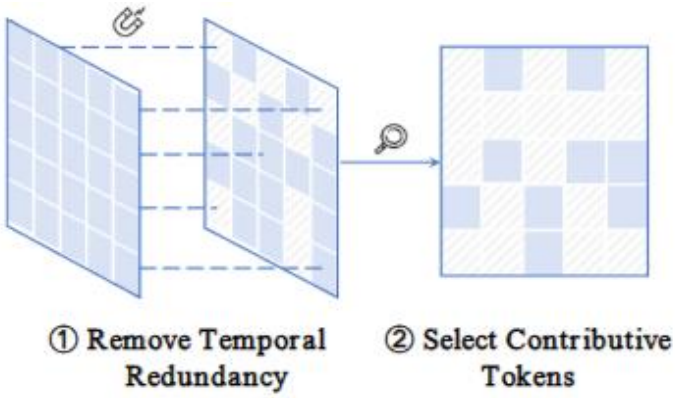
- Unified Spatiotemporal Token Compression for Video-LLMs at Ultra-Low Retention [CVPR 26]
  - Ultra-low retention을 위해 spatial·temporal redundancy를 통합적으로 고려
  - 제한된 token budget을 비디오의 시공간 정보에 효과적으로 배분하는 방법 제안

- KiToke: Kernel-based Interval-aware Token Compression for Video Large Language Models [ICML 26]
  - Low retention에서 발생하는 density/frequency 기반 selection bias를 완화
  - 특정 구간으로 token이 집중되는 것을 방지하고 temporal coverage를 보존

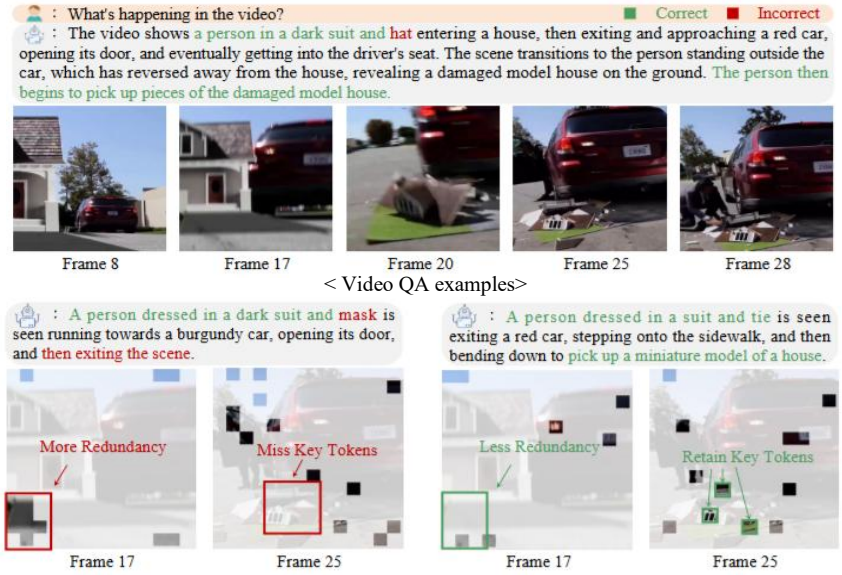
Unified Spatiotemporal Token Compression for Video-LLMs at Ultra-Low Retention [CVPR 26]

Introduction

- Stage-based video token compression
  - Video LLM은 많은 visual token으로 인한 높은 연산량을 요구
    - Spatial Compression 과 Temporal compression이 독립적으로 수행
      - 비디오 token간의 시공간 관계를 충분히 반영하지 못함
    - Ultra-low retention에서는 단계별 압축으로 인한 정보 손실과 token allocation이 어려움
    - 전체 spatio-temporal token을 하나의 global budget에서 통합적으로 선택할 필요가 있음

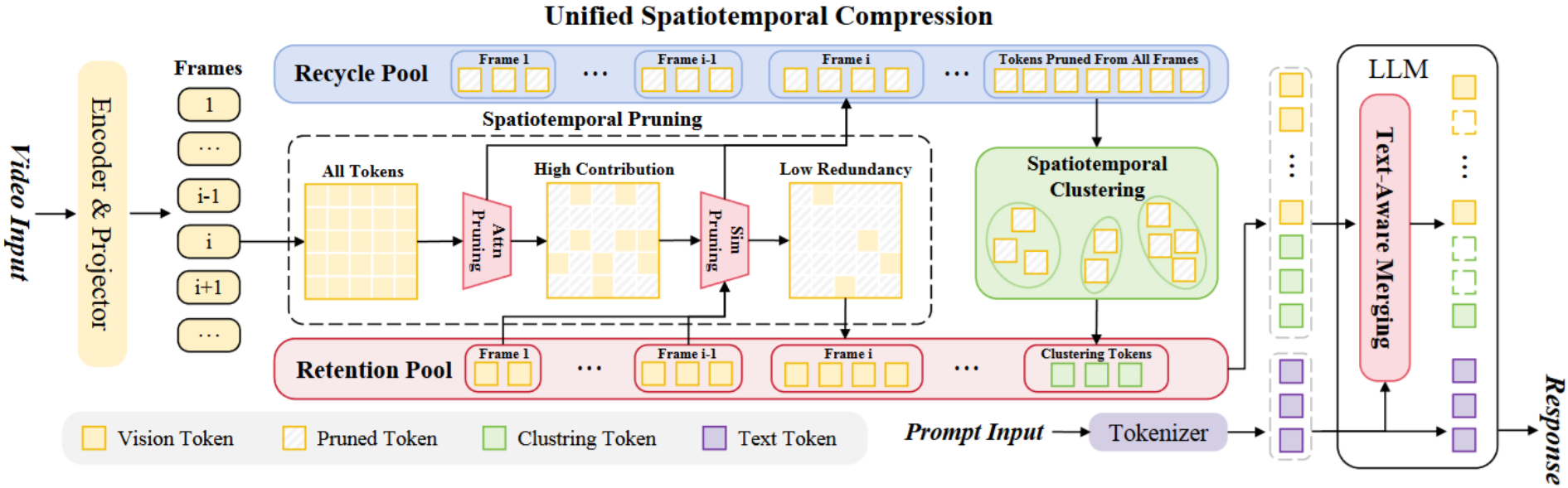


< Stage-based token compression framework >



Method

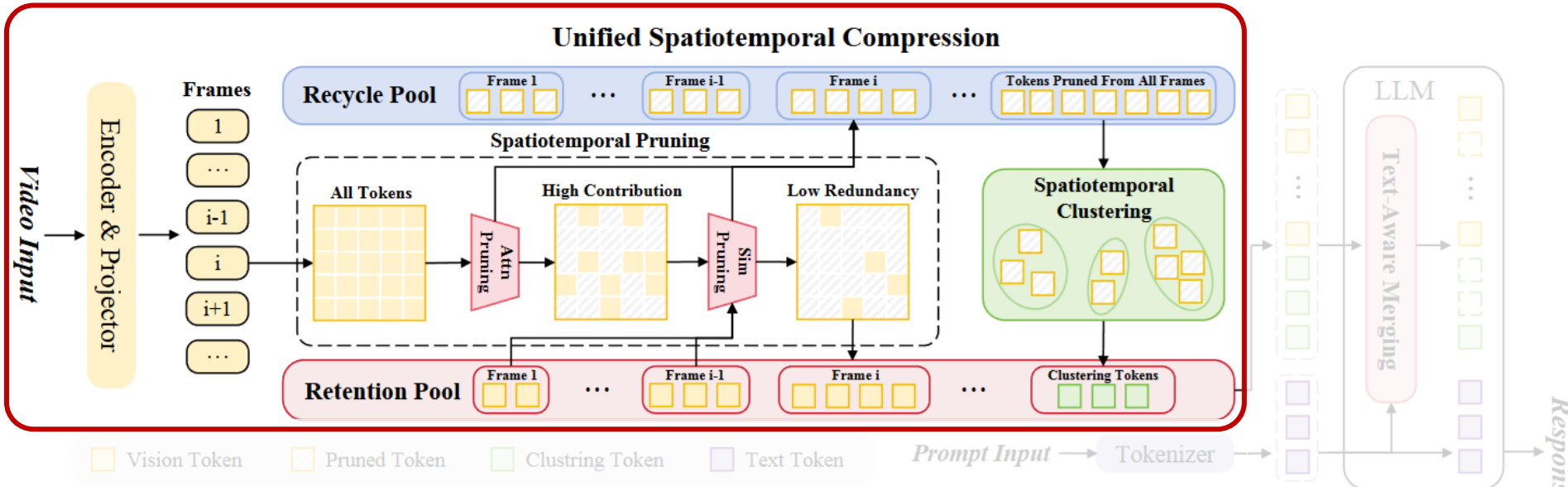
- Unified Spatio-Temporal Compression
  - 전체 visual token을 하나의 global retention pool에서 통합적으로 평가
  - Attention contribution과 semantic similarity로 중요하고 비중복적인 token을 선택
  - 제거된 token은 clustering 기반 merging과 refill을 통해 정보 손실을 보완
  - LLM 내부에서는 text-aware merging으로 query 기반 추가 압축을 수행



< USTC framework >

Method

- Unified Spatio-Temporal Compression
  - Pre-LLM Token compression stage
    - Vision encoder의 CLS token과의 attention score를 기반으로 vision token의 중요도 판단
      - Frame idx가 낮은 순서대로 진행되며 선정된 token은 retention pool로 이동
    - 앞의 frame에서 선정된 token 과의 cosine similarity를 통해 token 간의 diversity 판단
    - 선정 되지 못한 token의 경우, recycle pool에서 DPC-KNN을 통해서 retention pool에 반영



< Pre-LLM Token compression stage >

Method

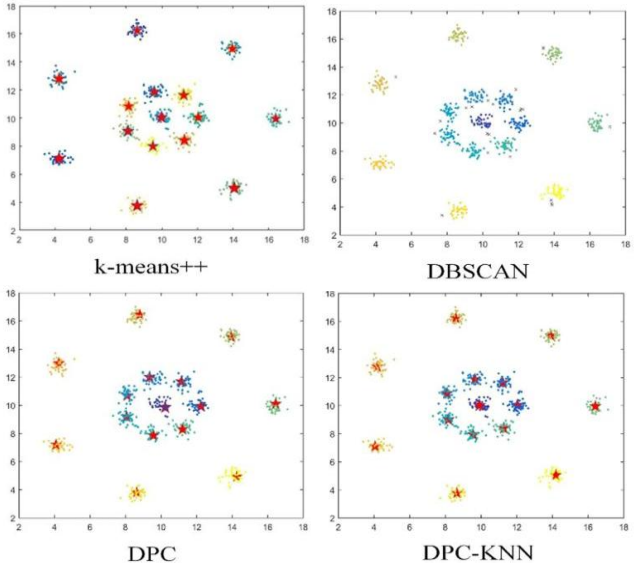
- Unified Spatio-Temporal Compression
  - DPC-KNN(Density Peak Clustering based on K-Nearest Neighbors)
    - 기존의 Density Peak Clustering 방법과 KNN 을 결합한 방식
    - KNN 거리로 각 token이 주변 token을 얼마나 잘 대표하는지 측정
    - Local density와 higher-density token까지의 거리를 함께 고려
    - 밀도가 높고 다른 고밀도 영역과 분리된 token을 cluster center로 선택
    - 불규칙한 token 분포에서도 의미적으로 구분되는 cluster를 형성하는 데 유리

$$\rho_i = \exp\left(-\frac{1}{k}\sum_{v_j \in \mathrm{kNN}(v_i)} d(v_i, v_j)^2\right),$$

$$\delta_i = \begin{cases} \max_{j \neq i} d(v_i, v_j) & \text{if } \rho_i = \max_k \rho_k \\ \min_{j:\rho_j > \rho_i} d(v_i, v_j) & \text{otherwise} \end{cases},$$

$$\text{scores } \gamma_i = \rho_i \times \delta_i$$

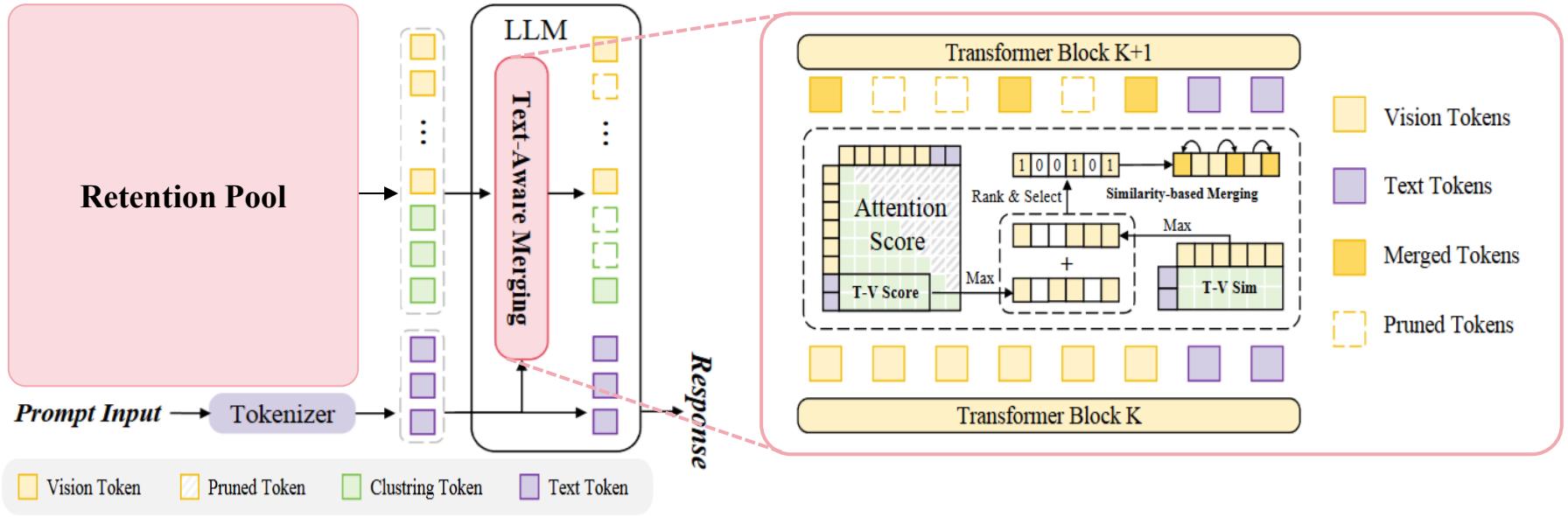
< Density peak clustering score >



< Clustering method comparison >

Method

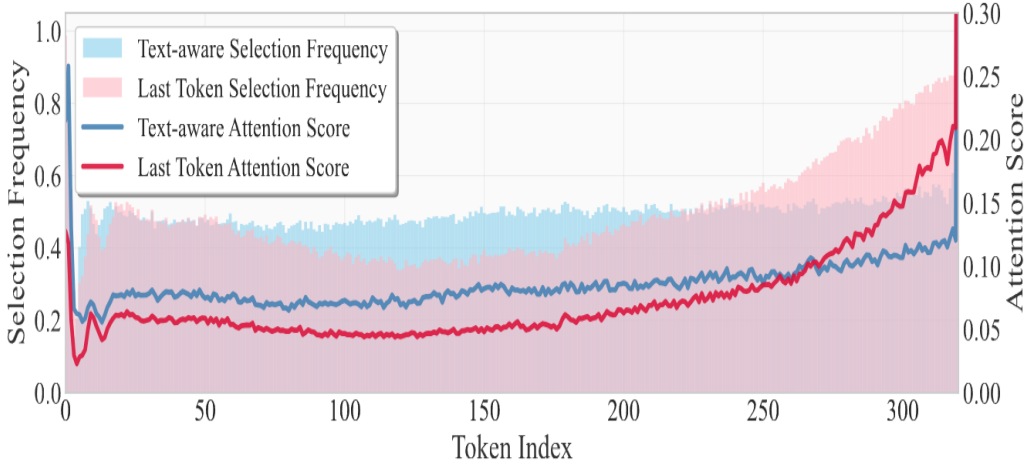
- Unified Spatio-Temporal Compression
  - Inner-LLM Token compression stage
    - Text-to-visual attention과 cosine similarity를 결합해 visual token의 query relevance를 평가
    - Attention 단독 사용 시 발생하는 RoPE 기반 relative position bias를 similarity score로 보완
    - Query relevance score 와 similarity score를 합해서 top-k로 retaining set을 선택
      - 나머지 token은 가장 유사한 retained token과 average merging



< Inner-LLM Token compression stage >

Method

- Unified Spatio-Temporal Compression
  - Inner-LLM Token compression stage
    - Bias from RoPE
      - RoPE는 query와 key를 Token 위치에 따라 회전시킨 다음 attention을 계산
      - Text Token과 sequenc에서 멀리 떨어진 경우, attention 점수가 낮아 지는 case 존재
        - ✓ 아래 그래프와 같이, sequence 상 마지막에 위치한 token의 선택율이 높음
      - 해당 문제를 완화 하기 위해서 Cosine similarity 를 같이 고려하여 score $I(\cdot)$로 사용



< Rotary Positional Embedding Bias visualization >

$$A_{qv} = A[N_v' :, : N_v'], \quad A_m = \max_{i=1}^{N_q} A_{qv}(i, :),$$

$$S_m(v_i) = \max_{t \in \mathcal{T}} \frac{v_i \cdot t}{\|v_i\| \|t\|},$$

$$A_m^{\text{norm}} = \frac{A_m - \min(A_m)}{\max(A_m) - \min(A_m)}.$$

$$S_m^{\text{norm}} = \frac{S_m - \min(S_m)}{\max(S_m) - \min(S_m)}.$$

$$I(v_i) = (1 - \lambda) \cdot A_m^{\text{norm}}(v_i) + \lambda \cdot S_m^{\text{norm}}(v_i).$$

< Final score formula>

Experiments

- Experiments Settings
  - Benchmark dataset
    - MVBench: 20개 비디오 이해 task와 4,000개 QA로 temporal comprehension을 평가
    - EgoSchema: 3분 길이의 1인칭 영상과 5지선다 QA로 장기적인 인간 행동 이해를 평가
    - MLVU: 3분에서 2시간 이상의 영상에서 추론·요약·세부 정보 검색 등 9개 task를 평가
    - LongVideoBench: 3,763개 영상과 6,678개 QA로 긴 영상의 understanding 능력을 평가
    - Video-MME: 11초부터 1시간 영상까지의 종합 비디오 이해 능력을 평가
  - Backbone model
    - LLava-onevision, LLava-video, Qwen-2.5-VL
  - Baseline model
    - Inner LLM method : FastV [ECCV 24]
    - Pre-LLM method : VisionZip [CVPR 25] , FastVid [NeurIPS 25]
    - Hybrid method : HoliTom [NeurIPS 25]

Experiments

- Comparison results
  - Ultra-low retention ratio ((5%, 2%, 1%))에서 기존 방법의 성능이 크게 저하
  - 제안 방법은 낮은 token budget에서도 중요한 시공간 정보를 효과적으로 보존
  - 모든 retention ratio에서 기존 압축 방법보다 일관되게 높은 성능을 달성

Method	FLOPs (T) ↓	FLOPs Ratio ↓	Retention Ratio	MVBench ↑	EgoSchema ↑	MLVU ↑	LongVideo Bench ↑	VideoMME ↑	Avg. ↑ Score	Avg. ↑ %
LLaVA-OV-7B [11]	41.4	100%	100%	58.3	60.4	47.7	56.4	58.6	56.3	100
FsatV [4]	7.9	19.1%	100%/10%	53.2	55.9	41.6	52.1	52.7	51.1	90.8
VisionZip [38]	4.0	9.6%	10%	53.5	58.0	42.5	49.3	53.4	51.3	91.2
LLaVA-Scissor [27]	4.0	9.6%	10%	-	57.5	-	-	55.8	-	-
FastVID [25]	4.0	9.6%	10%	55.9	58.7	42.6	56.3	**57.3**	54.2	96.2
HoliTom [23]	3.4	8.2%	10%/5%	57.3	**61.2**	45.1	56.3	56.8	55.3	98.3
Ours (w/o M)	4.0	9.6%	10%	57.4	60.5	44.7	**57.4**	56.6	55.3	98.3
Ours	3.4	8.2%	10%/5%	**57.7**	60.6	**45.5**	56.4	56.6	**55.4**	**98.4**
FastV [4]	6.4	15.5%	100%/5%	51.2	53.9	35.8	47.9	49.7	47.7	84.7
VisionZip [38]	2.2	5.4%	5%	45.2	51.9	37.4	46.4	48.2	45.8	81.4
LLaVA-Scissor [27]	2.2	5.4%	5%	-	56.6	-	-	53.3	-	-
FastVID [25]	2.2	5.4%	5%	53.0	57.1	42.1	51.1	54.2	51.5	91.5
HoliTom [23]	2.0	4.7%	5%/2.5%	55.6	**60.5**	40.6	53.6	54.2	52.9	94.0
Ours (w/o M)	2.2	5.4%	5%	**56.4**	59.5	42.5	53.7	**55.0**	53.4	94.9
Ours	2.0	4.7%	5%/2.5%	**56.4**	60.2	**42.8**	**54.5**	54.7	**53.7**	**95.4**
FastV [4]	5.5	13.3%	100%/2%	49.0	50.6	34.1	47.1	47.3	45.6	81.0
VisionZip [38]	1.2	2.9%	2%	41.7	47.6	31.8	45.1	45.9	42.4	75.3
FastVID [25]	1.2	2.9%	2%	48.0	52.3	37.6	47.3	49.2	46.9	83.3
HoliTom [23]	1.1	2.6%	2%/1%	52.6	57.2	37.4	48.5	51.1	49.4	87.7
Ours (w/o M)	1.2	2.9%	2%	**52.9**	57.2	39.5	**51.0**	51.3	50.4	89.5
Ours	1.1	2.6%	2%/1%	52.8	**57.6**	**40.3**	50.8	**51.8**	**50.7**	**90.1**
FastV [4]	5.2	12.6%	100%/1%	48.2	48.8	32.3	45.5	46.2	44.2	78.5
VisionZip [38]	0.9	2.1%	1%	40.8	43.8	29.7	44.4	44.3	40.6	72.1
FastVID [25]	0.9	2.1%	1%	45.3	47.8	32.4	46.1	47.0	43.7	77.7
HoliTom [23]	0.8	2.0%	1%/0.5%	49.6	52.9	33.9	48.1	49.0	46.7	82.9
Ours (w/o M)	0.9	2.1%	1%	**50.5**	53.3	**34.4**	**49.1**	**49.8**	**47.4**	**84.2**
Ours	0.8	2.0%	1%/0.5%	**50.5**	**53.8**	**34.4**	48.8	49.2	47.3	84.1

< LLava-OneVision Results>

Method	FLOPs (T) ↓	FLOPs Ratio ↓	Retention Ratio	MVBench ↑	EgoSchema ↑	MLVU ↑	LongVideo Bench ↑	VideoMME ↑	Avg. ↑ Score	Avg. ↑ %
LLaVA-Video-7B [46]	80.9	100%	100%	60.4	57.2	53.3	58.9	64.3	58.8	100
FsatV [4]	10.2	12.6%	100%/2%	42.9	41.0	34.3	45.6	48.1	42.4	72.1
VisionZip [38]	1.5	1.9%	2%	43.8	39.3	33.5	45.8	48.4	42.2	71.8
FastVid [25]	1.5	1.9%	2%	48.3	45.7	37.3	49.3	53.6	46.8	79.6
HoliTom [23]	1.4	1.7%	2%/1%	**50.2**	46.5	39.9	**50.7**	55.3	48.5	82.5
Ours (w/o M)	1.5	1.9%	2%	49.8	46.7	**40.8**	49.8	55.8	48.6	82.7
Ours	1.4	1.7%	2%/1%	50.1	**46.8**	**40.8**	50.2	**56.2**	**48.8**	**83.0**
FsatV [4]	9.7	11.9%	100%/1%	42.2	38.8	31.1	45.0	46.0	40.6	69.1
VisionZip [38]	1.2	1.5%	1%	42.3	36.8	30.9	45.6	47.0	40.5	68.9
FastVid [25]	1.2	1.5%	1%	43.6	38.8	30.1	46.2	49.3	41.6	70.7
HoliTom [23]	1.1	1.4%	1%/0.5%	46.4	41.1	38.9	48.8	51.7	45.4	77.2
Ours (w/o M)	1.2	1.5%	1%	47.9	**44.8**	40.0	48.9	54.6	47.2	80.3
Ours	1.1	1.4%	1%/0.5%	**48.0**	**44.8**	**40.1**	**49.4**	**54.9**	**47.4**	**80.6**
LLaVA-OV-0.5B [11]	3.54	100%	100%	46.6	26.6	33.5	47.5	43.7	39.6	100
FastV [4]	0.50	14.1%	100%/2%	39.2	21.2	22.5	38.8	34.0	31.1	78.5
VisionZip [38]	0.07	1.9%	2%	36.3	20.0	22.6	38.4	34.6	30.4	76.8
FastVid [25]	0.07	1.9%	2%	37.9	22.9	23.5	40.2	36.9	32.3	81.6
HoliTom [23]	0.06	1.8%	2%/1%	42.6	24.5	26.1	42.4	39.4	35.0	88.4
Ours (w/o M)	0.07	1.9%	2%	**43.0**	24.6	**26.4**	**45.3**	**40.5**	**35.9**	**90.7**
Ours	0.06	1.8%	2%/1%	42.7	**24.8**	**26.4**	45.2	40.4	**35.9**	**90.7**
FastV [4]	0.48	13.7%	100%/1%	35.6	19.3	21.5	37.9	32.2	29.3	74.0
VisionZip [38]	0.05	1.3%	1%	35.3	19.1	19.1	39.0	33.3	29.2	73.7
FastVid [25]	0.05	1.3%	1%	36.2	20.8	20.8	40.6	35.0	30.7	77.5
HoliTom [23]	0.05	1.3%	1%/0.5%	41.5	23.0	**25.1**	42.5	37.5	33.9	85.6
Ours (w/o M)	0.05	1.3%	1%	41.8	23.2	23.2	43.4	38.2	34.0	85.9
Ours	0.05	1.3%	1%/0.5%	**42.2**	**24.2**	23.1	**44.1**	**40.6**	**34.8**	**87.9**

< LLava-Video Results >

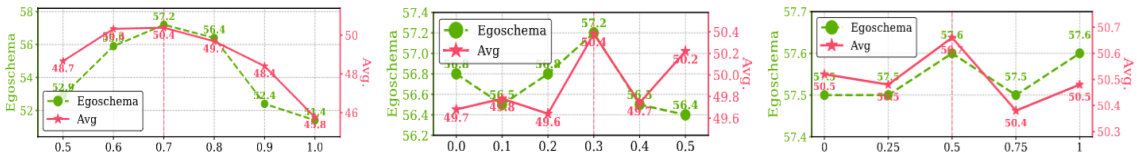
Method	Retention Ratio	MVBench	EgoSchema	MLVU	LongVideo Bench	VideoMME	Avg. ↑ Score	Avg. ↑ %
Qwen2.5-VL [2]	100%	63.0	52.8	38.5	55.0	57.5	53.4	100.0
VisionZip [38]	2%	50.3	45.3	29.7	43.4	44.1	42.6	79.8
FastVID [25]	2%	51.8	46.2	30.6	45.0	46.4	44.0	82.5
Ours(w/o M)	2%	**54.1**	47.8	**32.7**	45.4	**48.1**	**45.6**	**85.5**
Ours	2%	53.7	**48.4**	31.8	**45.6**	**48.1**	45.5	85.3
VisionZip [38]	1%	46.1	42.3	25.8	41.8	41.4	39.5	74.0
FastVID [25]	1%	47.2	43.5	26.4	**45.1**	43.1	41.1	76.9
Ours(w/o M)	1%	50.9	45.0	28.3	43.2	44.4	42.4	79.4
Ours	1%	**51.0**	**45.5**	**28.6**	43.4	**44.5**	**42.6**	**79.8**

< Qwen2.5-VL Results >

Experiments

- Ablation study
  - Spatio-Temporal Compression
    - Attention, similarity, clustering 모듈을 모두 적용했을 때 가장 높은 성능을 달성
      - Similarity-based pruning은 token 간의 중복을 줄여 성능을 크게 향상
      - DPC-KNN merging은 제거 대상 token의 정보를 보존하여 성능 향상에 기여
  - Frame length & Efficiency results
    - Frame 수가 늘면서 FLOPs를 줄임과 동시에 VideoMME 성능까지 향상 됨을 보임
    - Prefill 시간은 줄였으나, prosessing 시간이 여전히 큰 것을 확인

Attn	Sim	Cluster	MVBench	EgoSchema	MLVU	LongVideo Bench	VideoMME	Avg. ↑ Score	Avg. ↑ %
✓			41.4	46.6	30.8	44.7	45.3	41.8	74.2
	✓		51.4	55.1	38.8	50.0	50.9	49.2	87.5
		✓	52.2	55.0	36.3	48.4	51.3	48.6	86.4
✓	✓		52.1	56.8	38.4	49.9	51.2	49.7	88.2
✓		✓	48.3	51.4	33.2	46.4	49.5	45.8	81.3
	✓	✓	51.8	55.5	38.4	50.6	51.0	49.5	87.9
✓	✓	✓	**52.9**	**57.2**	**39.5**	**51.0**	**51.3**	**50.4**	**89.5**



(a) Ablation on τ. (b) Ablation on Clustering Token Ratio. (c) Ablation on λ.

< Ablation Study>

Method	Frames	FLOPs (T)	VideoMME Short	VideoMME Medium	VideoMME Long	VideoMME Total
Vanilla [11]	16	19.0	68.2	53.8	48.2	56.7
HoliTom [23]	64	1.7	63.0	53.9	46.3	54.4
Ours	64	1.7	62.4	53.3	47.6	54.5
HoliTom [23]	96	2.2	64.1	52.4	45.9	54.1
Ours	96	2.2	65.8	55.0	46.8	55.9
HoliTom [23]	128	2.8	66.2	53.2	46.8	55.4
Ours	128	2.8	**68.4**	**55.3**	**50.2**	**58.0**

< Frame length ablation >

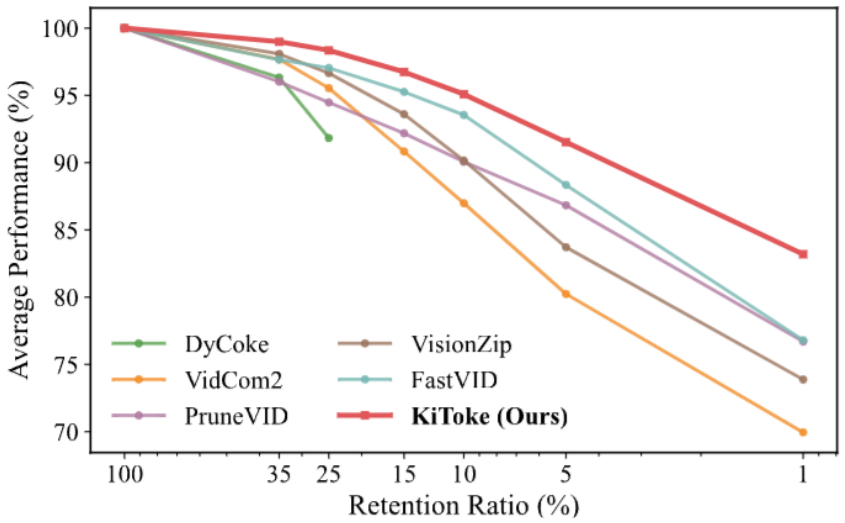
Method	Process	Prefill	TTFT	Throughput	Score
Vanilla [11]	-	701.2 (1×)	1104.7	27.6	56.3
VisionZip [38]	35.8	42.1 (17×)	451.6	35.4	45.8
FastVid [25]	**8.6**	43.1 (16×)	**446.2**	33.8	46.9
HoliTom [23]	88.7	40.5 (17×)	497.4	34.3	49.4
Ours	74.0	**31.3 (22×)**	473.8	**35.7**	**50.7**

< Efficiency Result>

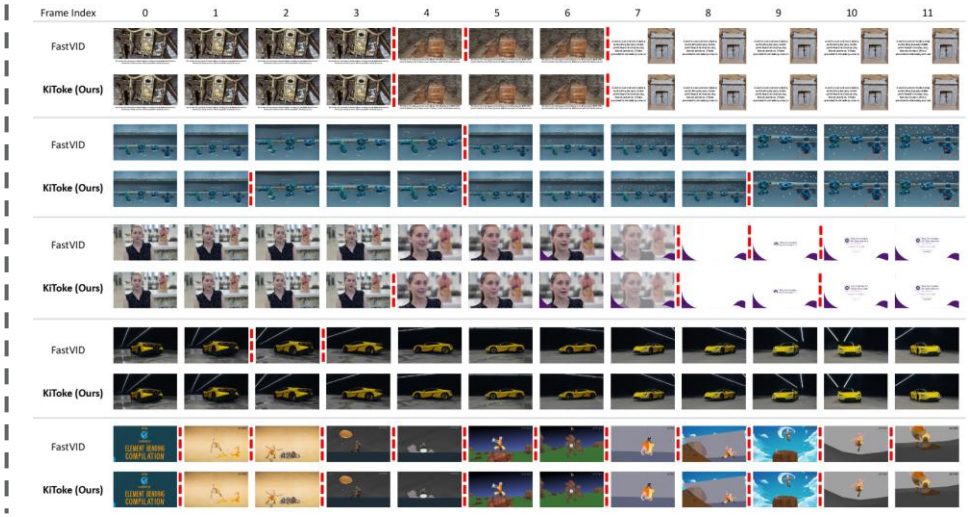
KiToke: Kernel-based Interval-aware Token Compression for Video Large Language Models [Arxiv 26]

Introduction

- 기존 방법의 한계점
  - Extreme token budget case에서 기존 방법의 경우 성능이 크게 저하
    - 원인으로는 global redundancy 와 density based selection bias를 지목
  - Local redundancy estimation
    - Video token에서 local window 또는 Temporal Segment 단위로 redundancy를 분석
      - 전체 Video에서 중복되는 부분에 대해서 다루지 못하는 한계가 존재
      - Local boundary의 quality에 의해서 성능이 좌우 되는 문제가 존재



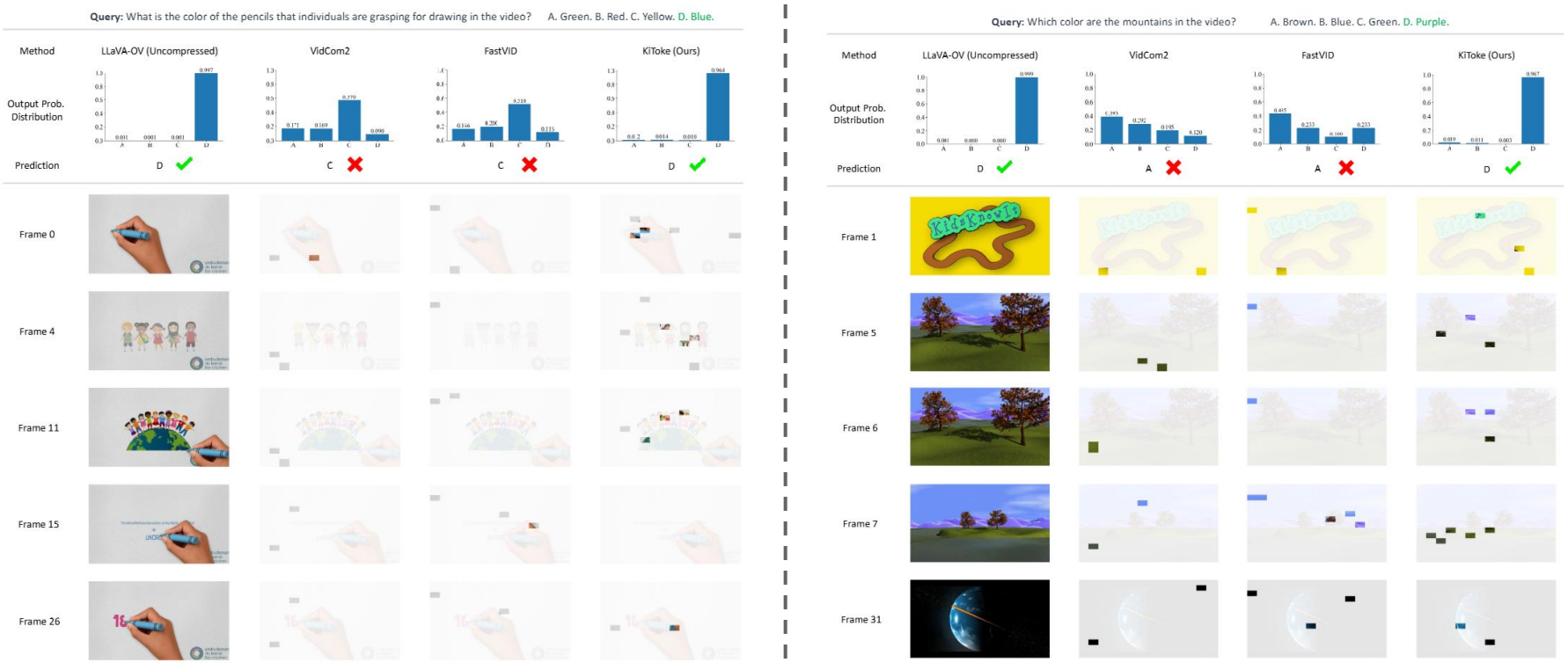
< Failure case of recent study on low retention budget >



< Failure case of local boundary selection>

Introduction

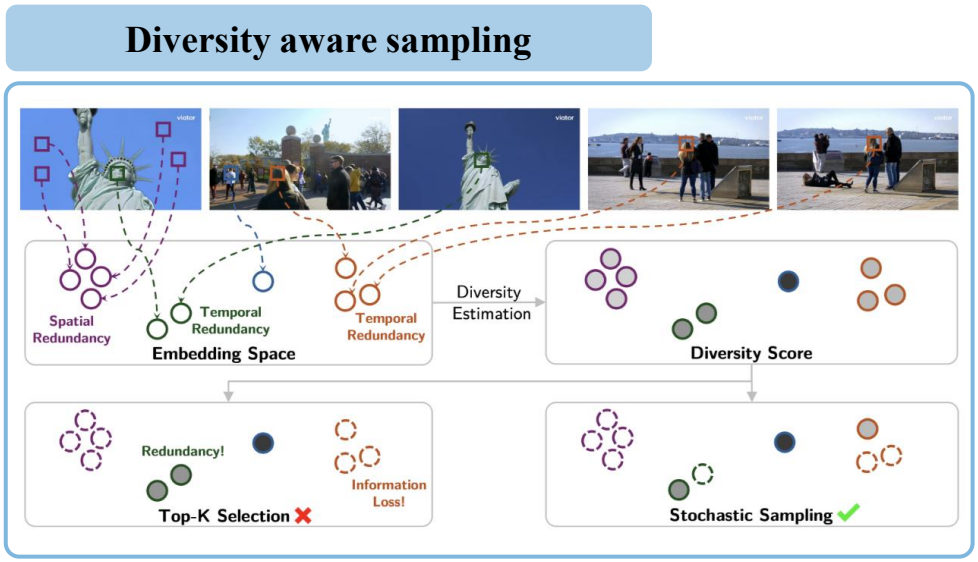
- 기존 방법의 한계점
  - Density / Frequency Bias
    - 기존 density 기반 방법은 자주 등장하는 token을 우선적으로 선택
    - 반복되는 배경이나 유사한 프레임에서 token이 주로 뽑히는 문제가 존재

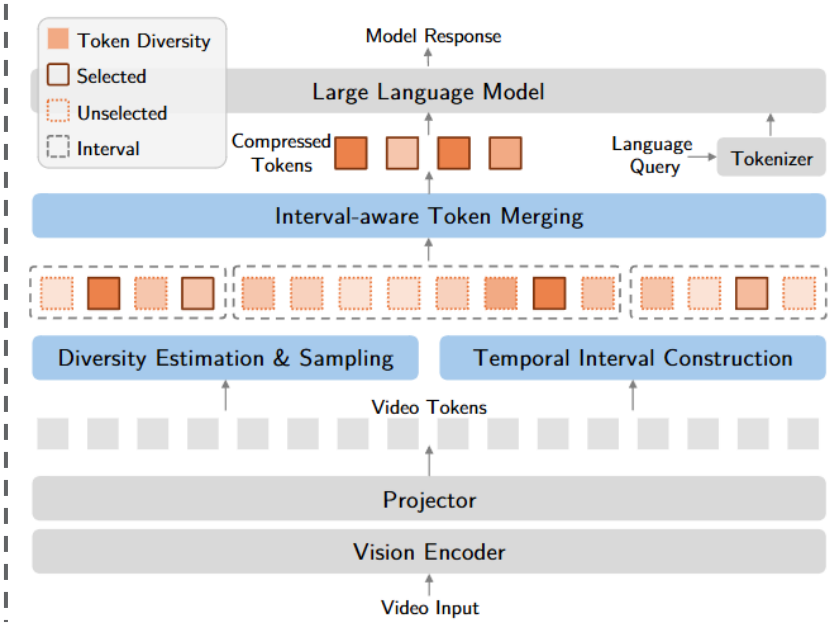


< Density / Frequency bias on recent study >

Method

- KiToke
  - Kernel density estimatio을 통해서 전체 token 각각에 대해서 diversity 계산
  - Frame 간의 appearance와 motion 변화를 통해 Temporal interval 결정
  - Diversity-aware token sampling을 통해서 Density에 의한 Bias 완화





< KiToke framework>

Method

- Kernel-based Diversity Estimation
  - 관측된 데이터 주변에 kernel을 배치하여 전체 데이터의 density를 추정하는 방법
    - 가까운 Data의 영향을 크게하고, 멀리 떨어진 데이터의 영향은 작게 반영
  - 여러 Kernel의 값을 합하여 각 위치의 데이터의 밀집 정도를 계산
    - 각 data point 위에 gaussian kernel을 놓고 이를 합쳐서 부드러운 density curve를 형성
  - 별도의 분포 가정 없이 부드러운 확률 밀도 함수를 추정하는 non-parametric 방법

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right)$$

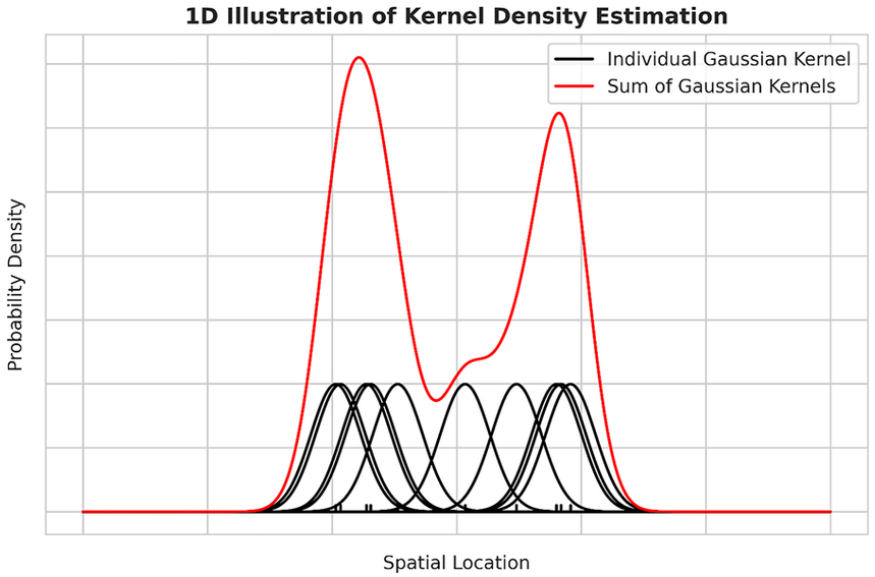
< Gaussian Kernel >

$$\hat{p}_h(x) = \frac{1}{Nh\sqrt{2\pi}} \sum_{j=1}^{N} \exp\left(-\frac{(x-x_j)^2}{2h^2}\right) \quad , x_j \sim \mathcal{N}(x_j, h^2)$$

< Probability density function of each point >

$$\hat{p}_h(x) = \frac{1}{N} \sum_{j=1}^{N} \mathcal{N}(x; x_j, h^2)$$

< Density curve function >



< 1D kernal density estimation visualization >

Method

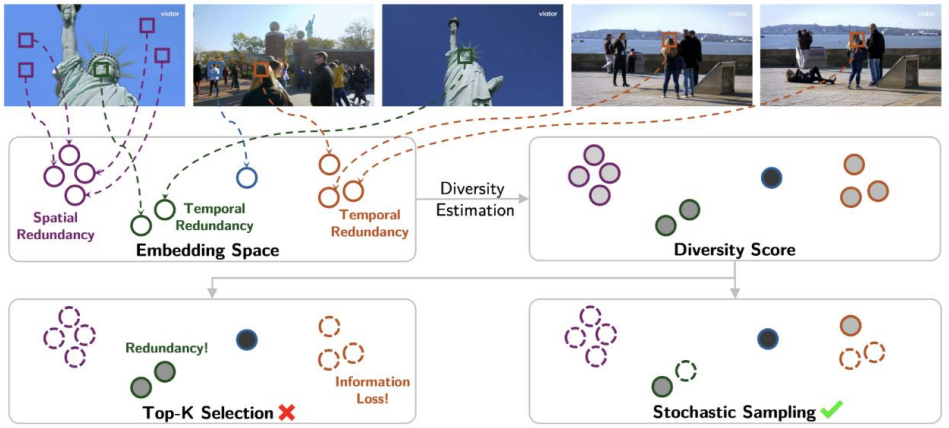
- Kernel-based Diversity Estimation
  - Token 간 상대 밀도 및 중복성을 측정하는 것이 목적
    - 정확한 확률 밀도를 측정하지 않아도 되니, normalize 단계는 생략하여 사용
    - KDE 와 동일한 방법으로 density curve 형성 이후 density의 역수를 diversity score로 사용
      - ∴ High Density → High Redundancy → Low Diversity Score
  - Top-k 방식이 아닌 diversity-weight sampling을 통해 density bias를 완화
    - Score를 확률로 하여 Stochastic Sampling을 진행
      - ∴ 희귀한 token의 경우 Score가 높기에, 선택될 가능성이 더 높아짐

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\alpha}\right)$$

< KiToke kernel function>

$$D_i = \sum_{j=1}^{N} \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\alpha}\right), \quad S_i = \frac{1}{D_i}$$

< KiToke density curve function >



< Density aware sampling >

Method

- Temporal Interval Construction
  - Frame 간의 변화가 달라지는 지점을 찾아 내용에 맞는 구간을 선택하는 방법
    - Token-level scoring (appearance 변화 추정)
      - Position-aligned difference : adjacent frame에서 동일한 위치의 token간의 변화량 측정
      - Best-match difference : 가장 유사한 Token 을 matching하여 두 token 간의 변화량 측정
    - Deviation-based scoring (motion 변화 추정)
      - Absolute deviation : 현재 변화량과 주변의 변화량의 차이를 측정
      - Relation deviation : 주변 변화에 대해서 변화 비율을 계산

$$\mathrm{diff}_t = \mathrm{diff}_t^{\mathrm{pos}} + \mathrm{diff}_t^{\mathrm{match}},$$

< Token-level scoring >

$$\mathrm{diff}_t^{\mathrm{pos}} = \frac{1}{M}\sum_{i=1}^{M} \|\mathbf{v}_{t,i} - \mathbf{v}_{t-1,i}\|_2$$

< Position-aligned difference >

$$\mathrm{diff}_t^{\mathrm{match}} = \frac{1}{M}\sum_{i=1}^{M} \min_j \|\mathbf{v}_{t-1,i} - \mathbf{v}_{t,j}\|_2$$

< Best-match difference >

$$\Delta_t = \max\left(\mathrm{diff}_t - \mathrm{diff}_{t-1},\ \mathrm{diff}_t - \mathrm{diff}_{t+1}\right)$$

< Absolute deviation >

$$\Delta_t^{\%} = \max\left(\frac{\mathrm{diff}_t - \mathrm{diff}_{t-1}}{\mathrm{diff}_{t-1}},\ \frac{\mathrm{diff}_t - \mathrm{diff}_{t+1}}{\mathrm{diff}_{t+1}}\right)$$

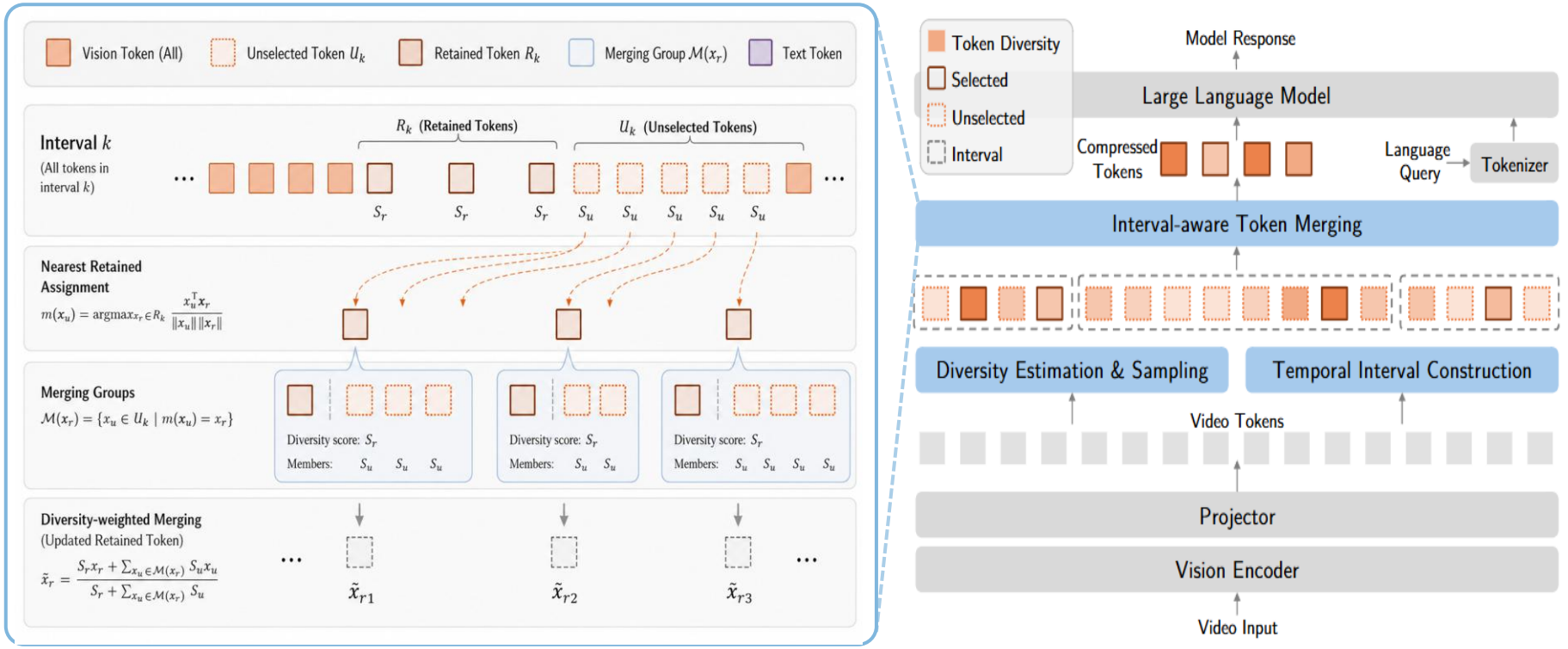
< Relation deviation >

$$\mathrm{diff}_t > \tau_{\mathrm{abs}} \quad \text{or} \quad \Delta_t > \tau_{\mathrm{dev}} \quad \text{and} \quad \Delta_t^{\%} > \tau_{\mathrm{rel}}$$

< Temporal interval construction algorithm >

Method

- Interval aware Token merging
  - 각 Temporal interval 내에서 unselected token의 정보를 보존 하기 위한 방법
    - Retained token과 unselected token 의 Cosine similarity 를 기반으로 top-1 matching
    - Matching 된 token의 경우, diversity-weighted averaging pooling을 진행



< Interval aware Token merging >

Experiments

- Experiments Settings
  - Benchmark dataset
    - MVBench: 20개 비디오 이해 task와 4,000개 QA로 temporal comprehension을 평가
    - MLVU: 3분에서 2시간 이상의 영상에서 추론·요약·세부 정보 검색 등 9개 task를 평가
    - LongVideoBench: 3,763개 영상과 6,678개 QA로 긴 영상의 understanding 능력을 평가
    - Video-MME: 11초부터 1시간 영상까지의 종합 비디오 이해 능력을 평가
  - Backbone model
    - LLava-onevision, LLava-video, Qwen3-VL
  - Baseline model
    - Inner LLM method
      - FastV [ECCV 24]
    - Pre-LLM method
      - VidCom^2 [ACL 25] , VisionZip[CVPR 25], PruneVID [CVPR25] FastVid [NeurIPS 25]

Experiments

- Comparison results
  - Ultra-low retention ratio에서 기존 방법의 성능이 크게 저하됨
  - KiToke는 global redundancy를 줄여 제한된 token budget을 효율적으로 활용
  - Diversity-weighted sampling과 interval-aware merging으로 temporal coverage를 보존
  - 대부분의 benchmark에서 기존 compression 방법보다 일관되게 높은 성능을 달성

Method	Retention Ratio γ	MVBench	LongVideo Bench	MLVU	VideoMME				Average	
					Overall	Short	Medium	Long	Score	%
Duration		16s	1~60min	3~120min	1~60min	1~3min	3~30min	30~60min		
LLaVA-OV-7B	100%	58.3	56.6	63.1	58.4	70.3	56.3	48.6	59.1	100%
FastV (Chen et al., 2024)	100%/25%	57.2	55.0	60.8	57.3	70.0	54.0	47.9	57.6	97.5%
DyCoke (Tao et al., 2025)	25%	57.4	53.8	58.4	53.9	64.1	52.7	44.8	55.9	94.6%
VidCom2 (Liu et al., 2025)	25%	57.0	55.3	62.4	**58.4**	69.1	57.0	**49.2**	58.3	98.6%
PruneVID (Huang et al., 2025)	25%	56.8	55.6	62.8	56.9	68.7	54.1	48.0	58.0	98.1%
VisionZip (Yang et al., 2025b)	25%	58.1	56.4	62.5	58.1	69.1	**57.3**	47.9	58.8	99.5%
FastVID (Shen et al., 2025)	25%	58.0	56.0	61.7	58.2	**70.3**	56.4	47.9	58.5	99.0%
KiToke (Ours)	25%	**58.4**	**57.4**	**63.4**	58.3	70.0	56.1	48.8	**59.4**	**100.5%**
FastV (Chen et al., 2024)	100%/10%	53.0	50.7	57.1	53.6	63.0	51.9	46.0	53.6	90.7%
VidCom2 (Liu et al., 2025)	10%	52.5	50.5	58.0	53.3	61.8	51.4	46.8	53.6	90.7%
PruneVID (Huang et al., 2025)	10%	55.7	54.5	60.7	56.2	66.9	53.6	**48.2**	56.8	96.1%
VisionZip (Yang et al., 2025b)	10%	55.6	52.7	59.4	55.6	64.9	54.6	47.4	55.8	94.4%
FastVID (Shen et al., 2025)	10%	**57.3**	55.5	61.1	**57.1**	**67.7**	**55.7**	47.9	57.7	97.6%
KiToke (Ours)	10%	57.2	**57.4**	**62.1**	56.0	66.3	54.4	47.1	**58.2**	**98.5%**
FastV (Chen et al., 2024)	100%/5%	49.9	48.3	55.4	51.5	58.0	50.9	45.7	51.3	86.8%
VidCom2 (Liu et al., 2025)	5%	46.9	47.4	53.6	49.4	55.7	49.4	43.1	49.3	83.4%
PruneVID (Huang et al., 2025)	5%	54.3	53.9	59.7	54.0	63.3	51.6	47.0	55.5	93.9%
VisionZip (Yang et al., 2025b)	5%	51.9	48.6	56.6	52.4	59.9	51.6	45.9	52.4	88.7%
FastVID (Shen et al., 2025)	5%	54.8	52.7	57.3	**54.6**	**64.6**	**53.0**	46.3	54.8	92.7%
KiToke (Ours)	5%	**55.2**	**54.9**	**60.8**	54.1	63.0	52.2	**47.1**	**56.2**	**95.1%**
FastV (Chen et al., 2024)	100%/1%	43.8	46.7	51.7	48.0	52.7	48.8	42.7	47.6	80.5%
VidCom2 (Liu et al., 2025)	1%	39.2	41.7	45.7	41.7	42.0	42.1	40.9	42.1	71.2%
PruneVID (Huang et al., 2025)	1%	46.0	46.7	53.7	48.0	52.0	47.7	**44.3**	48.6	82.2%
VisionZip (Yang et al., 2025b)	1%	42.6	43.9	52.1	46.5	49.6	47.3	42.7	46.3	78.3%
FastVID (Shen et al., 2025)	1%	45.7	46.1	51.9	46.9	52.3	46.4	41.8	47.6	80.5%
KiToke (Ours)	1%	**52.5**	**51.1**	**56.9**	**50.2**	**57.4**	**49.3**	43.8	**52.7**	**89.2%**

< LLava-OneVision Results >

Method	Retention Ratio γ	MVBench	LongVideo Bench	MLVU	VideoMME				Average	
					Overall	Short	Medium	Long	Score	%
Duration		16s	1~60min	3~120min	1~60min	1~3min	3~30min	30~60min		
LLaVA-Video-7B	100%	60.4	58.7	67.4	64.2	77.2	62.2	53.1	62.7	100%
FastV (Chen et al., 2024)	100%/25%	57.8	57.4	65.5	62.5	73.2	61.7	52.7	60.8	97.0%
DyCoke (Tao et al., 2025)	25%	56.6	56.2	56.8	59.4	72.3	56.0	49.8	57.2	91.2%
VidCom2 (Liu et al., 2025)	25%	57.0	56.5	58.6	61.4	73.2	60.7	50.2	58.4	93.1%
PruneVID (Huang et al., 2025)	25%	56.2	57.7	63.7	59.9	72.4	56.9	50.3	59.4	94.7%
VisionZip (Yang et al., 2025b)	25%	56.4	56.4	64.2	61.4	73.8	59.4	51.1	59.6	95.1%
FastVID (Shen et al., 2025)	25%	**59.2**	57.3	64.5	**63.7**	**75.2**	**62.6**	**53.4**	61.2	97.6%
KiToke (Ours)	25%	58.7	**58.2**	**66.5**	62.1	74.6	60.8	51.0	**61.4**	**97.9%**
FastV (Chen et al., 2024)	100%/10%	51.4	53.1	61.8	59.2	69.4	58.2	50.0	56.4	90.0%
VidCom2 (Liu et al., 2025)	10%	50.8	50.7	54.8	56.9	64.6	57.0	49.2	53.3	85.0%
PruneVID (Huang et al., 2025)	10%	54.1	55.2	61.0	57.3	68.9	54.7	48.2	56.9	90.7%
VisionZip (Yang et al., 2025b)	10%	55.0	52.1	58.3	59.0	69.7	57.4	49.8	56.1	89.5%
FastVID (Shen et al., 2025)	10%	57.1	55.7	61.6	**60.6**	**71.2**	**59.3**	**51.3**	58.8	93.8%
KiToke (Ours)	10%	**57.2**	**57.0**	**65.1**	59.1	70.7	57.7	49.1	**59.6**	**95.1%**
FastV (Chen et al., 2024)	100%/5%	49.0	50.3	57.7	55.7	64.4	55.4	47.3	53.2	84.8%
VidCom2 (Liu et al., 2025)	5%	46.5	47.0	52.5	52.8	57.9	53.4	47.1	49.7	79.3%
PruneVID (Huang et al., 2025)	5%	52.7	52.5	58.3	55.3	66.3	52.2	47.4	54.7	87.2%
VisionZip (Yang et al., 2025b)	5%	53.1	48.1	51.8	56.7	64.9	56.8	**48.4**	52.4	83.6%
FastVID (Shen et al., 2025)	5%	55.0	51.8	58.0	**58.0**	68.3	**57.3**	**48.4**	55.7	88.8%
KiToke (Ours)	5%	**55.6**	**55.7**	**62.5**	57.4	**69.3**	54.9	48.0	**57.8**	**92.2%**
FastV (Chen et al., 2024)	100%/1%	46.7	46.1	51.3	51.0	56.3	51.1	**45.7**	48.8	77.8%
VidCom2 (Liu et al., 2025)	1%	41.9	44.4	49.1	45.6	46.9	47.4	42.4	45.2	72.1%
PruneVID (Huang et al., 2025)	1%	44.0	47.4	52.8	49.6	53.8	49.7	45.2	48.5	77.4%
VisionZip (Yang et al., 2025b)	1%	43.3	45.9	50.6	49.7	53.4	51.0	44.6	47.4	75.6%
FastVID (Shen et al., 2025)	1%	45.0	48.2	52.0	51.1	58.0	50.3	45.0	49.1	78.3%
KiToke (Ours)	1%	**49.0**	**51.9**	**55.8**	**52.3**	**60.0**	**51.3**	**45.7**	**52.3**	**83.4%**

< LLava-Video Results >

Experiments

- Comparison results
  - Qwen 계열 모델에서도 llava 계열과 비교하여 동일한 결과 양상을 보임
  - 동일한 retention ratio에서 방법별 token 수와 LLM FLOPs는 유사
  - KiToke는 가벼운 압축 과정으로 compression overhead를 최소화
  - 10% retention에서 6.8×, 1%에서는 12.5×의 total prefill speedup을 달성

Method	Retention Ratio γ	MVBench	LongVideo Bench	MLVU	VideoMME				Average	
					Overall	Short	Medium	Long	Score	%
Duration		16s	1~60min	3~120min	1~60min	1~3min	3~30min	30~60min		
Qwen3-VL-8B	100%	68.5	62.5	71.4	68.7	79.9	67.0	59.1	67.8	100%
DyCoke (Tao et al., 2025)	25%	59.1	57.3	62.9	64.0	75.4	62.3	54.2	60.8	89.7%
VidCom2 (Liu et al., 2025)	25%	64.1	60.3	66.3	**66.3**	76.2	**65.0**	**57.8**	64.3	94.8%
PruneVID (Huang et al., 2025)	25%	58.4	59.2	65.5	62.5	72.9	60.3	54.3	61.4	90.6%
VisionZip (Yang et al., 2025b)	25%	**65.9**	60.4	67.0	65.4	76.4	62.8	56.9	64.7	95.4%
FastVID (Shen et al., 2025)	25%	65.1	59.7	68.2	63.5	74.3	61.4	54.8	64.1	94.5%
KiToke (Ours)	25%	65.2	**61.5**	**69.5**	65.8	**77.6**	63.9	56.0	**65.5**	**96.6%**
VidCom2 (Liu et al., 2025)	10%	54.5	54.7	60.3	61.6	70.3	59.4	**54.9**	57.8	85.3%
PruneVID (Huang et al., 2025)	10%	53.8	55.4	60.0	56.8	65.6	53.6	51.3	56.5	83.3%
VisionZip (Yang et al., 2025b)	10%	58.2	54.5	63.1	59.0	67.3	57.6	52.2	58.7	86.6%
FastVID (Shen et al., 2025)	10%	59.7	56.6	64.5	61.0	71.1	58.8	53.1	60.5	89.2%
KiToke (Ours)	10%	**60.5**	**58.6**	**67.0**	**62.8**	**74.3**	**61.2**	52.9	**62.2**	**91.7%**
VidCom2 (Liu et al., 2025)	5%	49.2	50.7	55.0	56.7	63.6	54.9	51.8	52.9	78.0%
PruneVID (Huang et al., 2025)	5%	49.8	53.9	57.8	53.8	61.4	51.4	48.4	53.8	79.4%
VisionZip (Yang et al., 2025b)	5%	51.7	49.6	57.2	55.5	60.7	53.8	52.1	53.5	78.9%
FastVID (Shen et al., 2025)	5%	54.3	53.4	60.9	57.7	65.0	56.6	51.6	56.6	83.5%
KiToke (Ours)	5%	**57.1**	**56.7**	**63.7**	**59.9**	**69.8**	**56.7**	**53.3**	**59.3**	**87.5%**
VidCom2 (Liu et al., 2025)	1%	41.7	45.3	47.2	46.4	45.3	46.8	47.1	45.1	66.5%
PruneVID (Huang et al., 2025)	1%	42.6	48.1	52.1	48.4	52.1	47.2	45.8	47.8	70.5%
VisionZip (Yang et al., 2025b)	1%	41.4	45.8	49.4	47.1	48.2	46.2	46.8	45.9	67.7%
FastVID (Shen et al., 2025)	1%	44.0	46.5	54.0	49.5	53.8	48.8	46.0	48.5	71.5%
KiToke (Ours)	1%	**48.2**	**50.7**	**56.3**	**53.6**	**60.9**	**50.2**	**49.7**	**52.2**	**77.0%**

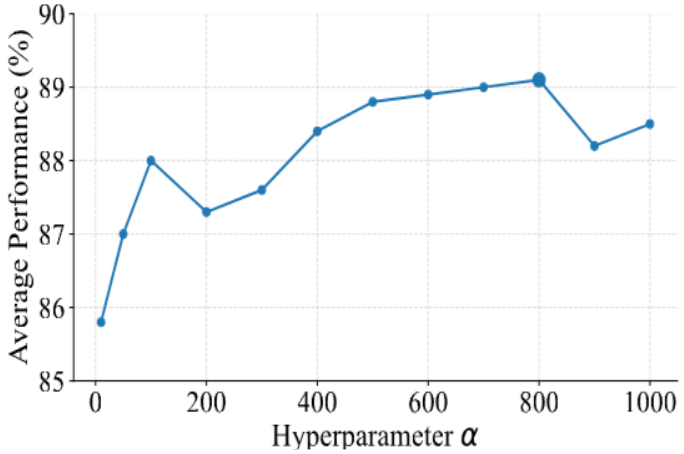
< Qwen3-VL Results >

Method	# Token	TFLOPs	Prefill Time (ms)			Avg. Acc.
			Compression	LLM Forward	Total	
Vanilla	6272	48.82	-	448.0	448.0 (1.0x)	59.1 (100%)
VidCom2	627.0	4.17	70.7	53.1	123.8 (3.6x)	53.6 (90.7%)
PruneVID	628.8	4.18	81.2	53.9	135.1 (3.3x)	56.8 (96.1%)
VisionZip	640.0	4.26	76.4	56.5	132.9 (3.4x)	55.8 (94.4%)
FastVID	639.8	4.26	15.3	56.1	71.4 (6.4x)	57.7 (97.6%)
KiToke (ours)	**627.0**	**4.17**	**13.1**	**53.1**	**66.2 (6.8x)**	**58.2 (98.5%)**
VidCom2	65.5	0.43	70.8	24.3	95.1 (4.7x)	42.1 (71.2%)
PruneVID	67.2	0.44	81.0	24.5	105.5 (4.2x)	48.6 (82.2%)
VisionZip	64.0	0.42	76.5	23.9	100.4 (4.5x)	46.3 (78.3%)
FastVID	62.9	0.41	15.2	23.7	38.9 (11.5x)	47.6 (80.5%)
KiToke (ours)	**62.0**	**0.41**	**12.3**	**23.6**	**35.9 (12.5x)**	**52.7 (89.2%)**

< Efficiency results>

Experiments

- Ablation study
  - Ablation study 결과 Merging 이 Pruning 방법보다 유리한 것을 보임
    - Merging 시에, Avaerage pooling을 하는 것보다 weighted pooling이 유리
  - Sampling method에 있어서, Stochastic sampling이 Top-k 방식과 비교하여 우수
    - Stochastic sampling method 의 경우 Pivotal sampling 방식이 유리
      - Multinomial sampling : 확률에 따라 token을 무작위로 선택
      - Pivotal sampling : token 간 확률을 조정하며 정확히 K개를 낮은 변동성으로 선택
  - Kernel의 분산이 커질수록 성능이 오르나 큰 분산의 경우, 오히려 성능이 저하



< Kernel parameter ablation study >

Row	Interval	Token Merge	MVBench	LongVB	MLVU	VideoMME	Average
1	✗	✗	51.9	47.3	53.6	48.6	50.3
2	✗	Uniform	51.6	49.1	55.6	49.5	51.4
3	✗	Weighted	51.9	49.4	56.0	49.9	51.8
4	Cluster	Weighted	51.8	49.7	56.3	49.7	51.9
5	Frame Sim.	Weighted	51.9	50.7	56.6	50.0	52.3
6	Ours	Uniform	52.3	49.8	55.8	50.1	52.0
7	Ours	Weighted	**52.5**	**51.1**	**56.9**	**50.2**	**52.7**

< Ablation study >

Method	MVBench	LongVB	MLVU	VideoMME	Average
Top-K	51.6	49.4	56.0	49.6	51.7
Multinomial Sampling	52.5 (±0.26)	49.7 (±0.51)	56.6 (±0.38)	50.2 (±0.31)	52.3 (±0.23)
Pivotal Sampling	**52.6 (±0.25)**	**51.1 (±0.25)**	**57.0 (±0.29)**	**50.3 (±0.29)**	**52.7 (±0.16)**

< Sampling method ablation study >

감사합니다.