Dealing with the Elephant in the KV Cache: Video Frame Sampling for Multimodal LLM Inference

Konstantinos Papaioannou, Thaleia Dimitra Doudali

IMDEA Software Institute

@ EuroMLSys '26

Edinburgh, April 27th

The **M**ultimodal **L**arge **L**anguage **M**odels Effect

Explain Images

User



Do you know who drew this painting?

Model

The painting depicts a woman commonly believed to be Mona Lisa, the famous artwork by Leonardo da Vinci.

Summarize Videos

User



Please describe this video.

Model

The video begins with a person holding a tabby cat, with the text ‘DOGS WITH BACK PAIN’ and a crying emoji displayed at the bottom of the screen. The person, who has tattoos on their arms and is wearing a light-colored shirt, is in a room with large windows and a white wall. The scene transitions to the person petting a large, brown and white dog, who is sitting on the floor...

Impact of Video on Workload Performance

Early Foundation Models



GPT-3

LLaMA

MISTRAL AI_

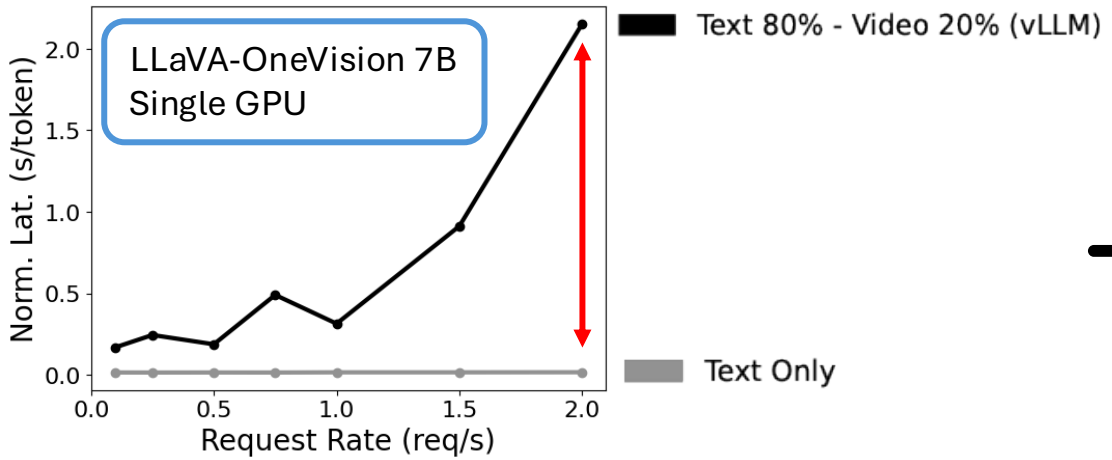
Emerging Multimodal LLMs

GPT-4

Gemini

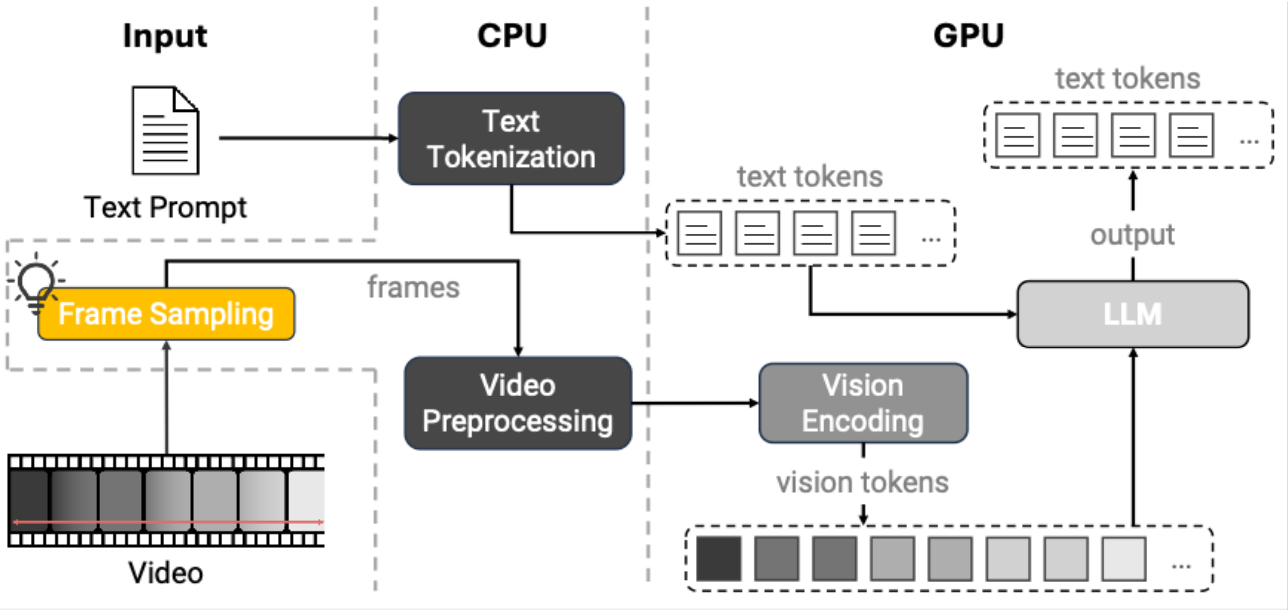
Claude

deepseek



Warning: Workloads with videos are much slower than text-only workloads !!!

MLLM Inference Overview



	Text ShareGPT	Video Video-MME
TTFT	0.03 s	0.87 s
Memory*	0.01 GB	0.67 GB
	~256 tokens	~12.5k tokens

*memory = #tokens in the KV Cache



Challenge: While text is light, videos act as an "**elephant**" in MLLM inference, taking **orders of magnitude** more memory and time to process.

Limitations of State-of-the-art MLLM Inference Systems

Category	System	Approach	Limitations
Resource Disaggregation	ModServe @ SoCC '25	Independent modality processing	**More resources needed** Efficient GPU interconnect required
Token Sparsification	SparseVLM @ ICML '25	Visual-tokens selection based on text-tokens	Semantic-dependent and potentially **decreased accuracy**



Current works treat videos as the "**elephant" in the room**. They acknowledge their impact, but neglect opportunities to truly reduce their overheads.



Our Approach: What if we *directly* reduce the video input itself?
How? Select a few informative frames instead of processing the full video.

Reduce Input Early

Use Fewer Resources

Preserve Model Accuracy

Experimental Setup – Initial Results

Testbed vLLM

1× NVIDIA A100 40GB GPU

Models

- **LLaVA-OneVision** 500M / 7B
- **Qwen2-VL** 2B / 7B
- **Pixtral** 12B

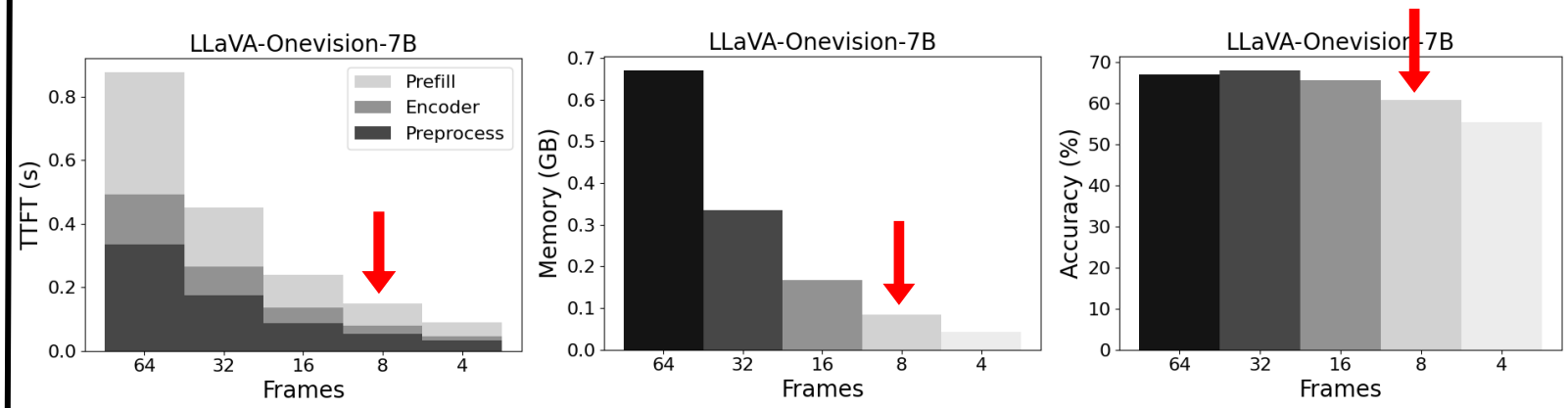
Benchmarks

- **Video-MME**: Multiple Choice
- **MMBench-Video**: Open-Ended Q&A
- **TempCompass**: Captioning

Metrics

TTFT (s) - Normalized Latency (s/tkn)
Memory Footprint (GB)
Accuracy (%) using LLM-as-a-judge

Idea: Uniformly processing *fewer frames* → inference acceleration



Half the frames lead to ***half*** the TTFT and memory, but accuracy drops a lot.

Can we *preserve accuracy* by selecting fewer frames in a *smarter* way?

*vLLM by default selects a constant number of frames (e.g. 64, 32,..) *regardless of the video duration.*

Frame Selection Techniques Comparison

Techniques

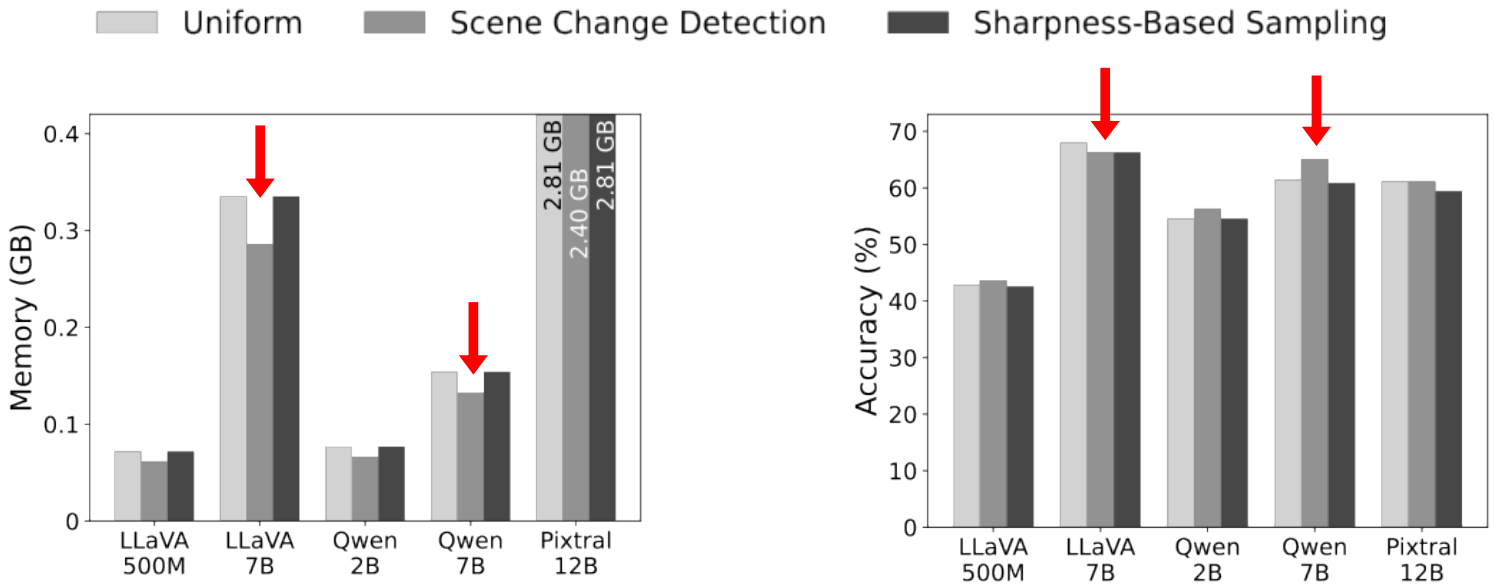
Uniform (vLLM):
fixed number of frames at regular intervals

Scene Change Detection:
selects frames at major visual transitions

Sharpness-Based Sampling:
selects high quality frames avoiding blurry ones

Use Case: Multiple Choice Q/A*

*more use cases in the paper



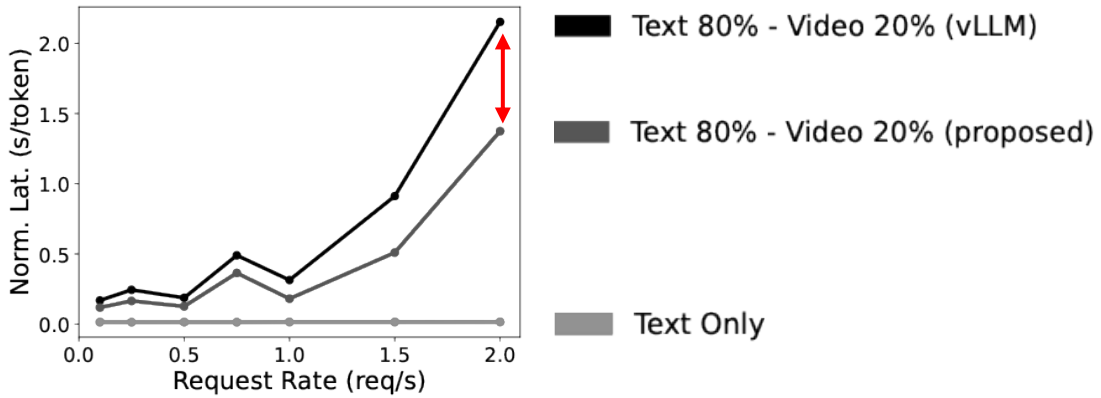
Takeaway: Smart frame selection techniques like **scene change detection** *reduce memory* for the same input budget *without hurting accuracy.*

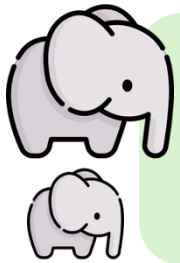
Our Proposed Approach

Reduced frame count + smart frame selection

↓

max. 32 frames + **S**cene **C**hange **D**etection





Takeaways: Our proposed approach **accelerates** MLLM inference by **1.57x** compared to vanilla vLLM, ***reducing the existing performance gap.***

This approach *directly addresses* the **elephant** in MLLM inference (videos) and *reduces* its impact.

🎯 Reduce Input Early 🎯 Use Fewer Resources 🎯 Preserve Model Accuracy

Paper

Future work: Develop a dynamic framework that adapts across model families and use cases while balancing accuracy guarantees against performance objectives.