DIVE-Bench: Codec-Style High-FPS Video Understanding

Haichao Zhang[1]*, Wenhao Chai[2], Shwai He[3], Ang Li[3], and Yun Fu[1]

[1] Northeastern University, Boston, MA, USA
[2] Princeton University, Princeton, NJ, USA
[3] University of Maryland, College Park, MD, USA
zhang.haich@northeastern.edu, wenhao.chai@princeton.edu,
{shwaihe,angliece}@umd.edu, yunfu@ece.neu.edu

DIVE-Bench: https://hai-chao-zhang.github.io/DenseVideoUnderstand/

Abstract. High temporal resolution is essential for capturing fine-grained dynamics in video understanding, yet current video large language models (VLLMs) and benchmarks predominantly rely on low-frame-rate sampling, discarding dense temporal information. This design compromise is driven by the high cost of frame-wise tokenization, which incurs redundant computation and linear growth in token count with video length. While effective for slowly changing content, such engineering trade-offs severely limit tasks where information is distributed across nearly every frame, such as educational or high-motion videos that require frame-by-frame reasoning. To address this gap, we introduce the novel task of **High-FPS Video Understanding**, the first task specifically designed to evaluate video comprehension at high frame rates, where information is present in nearly every frame. We further propose the first benchmark, **DIVE-Bench** (Dense Information Video Evaluation Benchmark), to expose the limitations of existing benchmarks whose question–answer pairs are not sensitive to dense temporal variations. To enable efficient high-FPS processing, we propose **Gated Residual Tokenization (GRT)**, a codec-style inter-tokenization framework that operates not only after but also during tokenization. First, *Motion-Compensated Gated Inter-Tokenization* applies pixel-level motion estimation to gate static regions during tokenization, achieving sub-linear growth in both tokenization time and token count. Second, *Semantic-Scene Token Merging* further reduces redundancy by merging semantically similar static tokens while preserving dynamic content. Extensive experiments on DIVE-Bench show that GRT improves high-FPS efficiency while preserving or improving answer quality under dense temporal perception tasks. These results highlight the importance of dense temporal modeling and establish codec-style inter-tokenization as a practical direction for scalable high-FPS video understanding.

* Corresponding author.

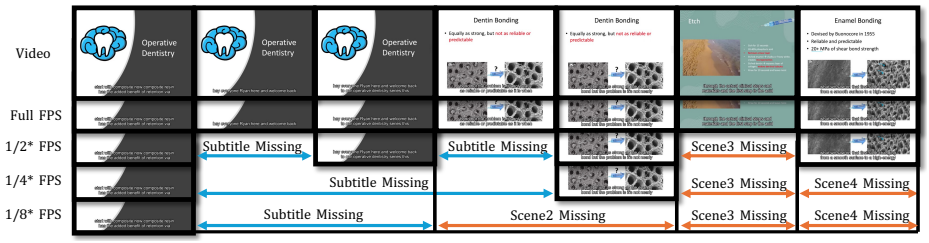

Fig. 1: Visualization on impact of a frame-rate reduction on video question answering. Most current models sample videos with a strict frame limit and at fixed time intervals, resulting in a very low effective frame rate. Here, we use subtitle reading (OCR [2]) in an educational video benchmark to illustrate the impact of frame rate reduction. Frame-by-frame analysis is required to capture all subtitles; when the frame rate is halved, some subtitles are missed and short scene segments become invisible to the video large language model. As the frame rate decreases further, more scenes are skipped. At one-eighth of the original rate, only one frame remains per segment, rendering frame-by-frame reasoning impossible.

1 Introduction

Does visual frame density matter for humans or for video large language models? The short answer is yes. Visual frame density, the number of frames captured per second, has long been fundamental to both biological and artificial vision systems, especially for physical AI systems that must perceive fine-grained temporal dynamics [28]. Many species rely on high-frame-rate perception to detect rapid changes essential for survival, while engineers invest heavily in high-speed camera technology [11] to capture fine-grained motion [3]. Likewise, audiences prefer high-frame-rate video and gaming content for smoother and more immersive experiences. Together, these observations highlight the importance of dense temporal sampling for accurate perception and information extraction. Such high-FPS cues are also central to physical AI problems such as trajectory forecasting, out-of-sight tracking, and world modeling [30–33], yet current VLLMs are rarely designed to preserve them under sparse frame sampling.

In video understanding research, large language models (LLMs) must interpret dynamic visual streams by converting frames into token sequences. However, most existing approaches drastically reduce the input frame rate, sampling only a handful of frames per second, and discard the majority of temporal information. As illustrated in Fig. 1, this coarse sampling can cause critical details, such as reading lecture subtitles or brief instructional segments to vanish. At extreme subsampling rates (one-eighth of the original), each segment may contain only a single frame: sufficient to convey a general scene, but wholly inadequate for tasks that require frame-by-frame reasoning.

Despite this, existing video large language models (VLLMs) [10, 16, 34, 36] typically operate on sparse frame subsets at low FPS, discarding dense temporal cues present in the omitted frames. Many methods impose a strict frame budget, effectively reducing FPS as video length increases. Notably, some approaches even report state-of-the-art results using as few as six frames per clip [6]. These

results indicate that current benchmarks do not require high temporal resolution to perform well—an observation that directly contradicts the qualitative evidence in Fig. 1 and undermines the motivation for fine-grained, high-FPS video understanding. This discrepancy motivates the need for a new task formulation that explicitly targets dense temporal reasoning.

We attribute this limitation in current VLLM research to three fundamental challenges. **First**, high-FPS videos generate an excessive number of tokens under standard patch-based tokenization, quickly overwhelming the limited throughput of existing LLMs [38]. Since self-attention scales quadratically with sequence length [29], inference becomes intractable as FPS increases. Without effective token reduction, no current video LLM can efficiently process full-FPS inputs. **Second**, widely used tokenizers such as CLIP [18] and SigLIP [27] rely on convolutional layers that must process entire images before patch tokenization. This design prevents parallelism across dense video frames and leads to linear growth in both tokenization time and token count as FPS increases. What is needed instead is a gated inter-tokenization mechanism that filters uninformative patches prior to embedding, enabling sub-linear computational growth. **Third**, existing video understanding benchmarks are designed around these model constraints. They primarily pose coarse-grained queries, such as activity recognition or object presence, that can be answered using only a few frames. As a result, they fail to evaluate dense temporal reasoning or stress-test high-FPS processing, reinforcing the need for a new benchmark for high-FPS video understanding.

To address these challenges, we make the following key contributions. First, we introduce the novel task of **High-FPS Video Understanding**, which aims to evaluate a model's ability to reason over densely sampled, high-frame-rate video content. To support this task, we propose **DIVE-Bench** (Dense Information Video Evaluation Benchmark) (Sec. 4), the first benchmark explicitly designed for high-FPS video question answering. DIVE-Bench consists of densely sampled video clips paired with QA tasks that require true frame-by-frame reasoning, where skipping even a small number of frames leads to immediate information loss (Fig. 1). Second, we present **Gated Residual Tokenization** (GRT), a codec-style framework for accelerating and reducing tokenization in high-FPS video settings. The first stage, Motion-Compensated Gated Inter-Tokenization, treats video tokenization analogously to key-frame and residual-frame coding: static regions are represented by key-frame tokens, while changed regions are encoded as residual P-frame tokens. This design filters uninformative patches before tokenization and achieves sub-linear growth in both tokenization time and token count as FPS increases (Sec. 3.2). Third, we introduce a Semantic-Scene Token Merging module that further compresses token sequences at the semantic level. After extracting key-frame and P-frame token sets, we compute distributional similarity across scenes to merge semantically redundant key tokens while preserving motion-specific P-frame tokens. This complementary step reduces sequence length while retaining the dense temporal cues needed for downstream reasoning (Sec. 3.3).

Our main contributions are listed as follows:

1. **High-FPS Video Understanding Task:** We introduce the first task explicitly targeting video understanding at high frame rates, overcoming the limitations of prior methods based on sparse frame sampling.
2. **DIVE-Bench:** We propose **DIVE-Bench**, the first high-FPS video QA benchmark that requires true frame-by-frame reasoning and reveals the inadequacy of coarse-temporal evaluations.
3. **GRT:** We present a codec-style gated residual tokenization framework that combines motion-compensated gated inter-tokenization with semantic-scene token merging, achieving sub-linear complexity while preserving dense temporal information.
4. **Empirical Validation:** Experiments on DIVE-Bench and existing video benchmarks show that GRT improves high-FPS efficiency and preserves or improves answer quality, demonstrating the scalability and practicality of codec-style inter-tokenization.

2 Related Work

2.1 Frame Sampling in VLLMs

Video large language models (VLLMs) extend vision–language pretraining to dynamic inputs, enabling tasks such as video question answering and captioning. This direction is closely related to broader studies on multimodal alignment and MLLM evaluation [13–15]. Early works like Flamingo [1] and MERLOT Reserve [26] demonstrated the feasibility of aligning video frames with text, while recent architectures such as LLaVA and its variants [10, 12] adapt large language models (LLMs) to video by inserting vision embeddings into transformer layers. Despite these advances, most VLLMs rely on sparse temporal sampling, i.e., selecting a fixed number of frames per clip. For example, LLaVA-One Vision limits clips to 32 frames, LLaVA-Vid to 20 frames, and VideoLLaVA to just 8 frames [6]. Such caps simplify computation but discard dense temporal cues, preventing models from performing frame-by-frame reasoning on high-FPS content. Recent works also emphasize dense temporal information for motion-language alignment [22,24], cognitive MLLM reasoning [9], and egocentric world modeling [19], but not for high-FPS VLLM evaluation.

2.2 Patch Tokenization and Frame Selection

Standard VLLM pipelines tokenize each selected frame into a grid of patch embeddings (e.g., 16x16 patches in ViT) before feeding them into the language model. Frame selection is typically based on uniform sampling or simple motion heuristics [23, 25], but these strategies still process every patch in each sampled frame, yielding long token sequences and high preprocessing latency. Adaptive sampling methods [6] allocate more frames to high-motion segments, yet they do not address the quadratic self-attention cost arising from large token budgets. Moreover, even flexible-FPS variants retain caps on total frames, limiting their ability to leverage very high frame rates.

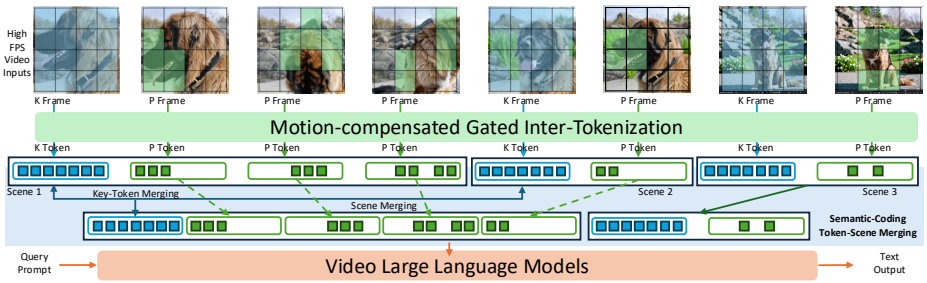

Fig. 2: Overall architecture of our codec-style gated residual tokenization pipeline. Given an input video, we first apply a pixel-coding, motion-compensated gated inter-tokenization process: key frames extract static scene tokens, while P-frames capture moving patches as residual P-token sets. Each scene thus yields one key-token set and multiple P-token sets. A subsequent semantic-coding token-scene merging module measures similarity between adjacent scenes (via their key-token distributions) and merges semantically equivalent key tokens while concatenating the P tokens. The resulting reduced token sequence, together with the query prompt tokens, is then fed into a video large-language model to generate the final answer.

3 Methodology

3.1 New Task: High-FPS Video Understanding

High-FPS video content is crucial for capturing fine-grained temporal dynamics. However, it has been largely overlooked in video understanding research and in video-LLM design, due to the lack of suitable benchmarks and the prohibitive length of resulting token sequences. We define *High-FPS Video Understanding* as the task of preserving and reasoning over a densely sampled, high-frame-rate visual stream under a practical token budget. Unlike conventional sparse-sampling strategies that collapse a video to a small fixed set of keyframes, this task keeps the temporal evidence needed for frame-by-frame reasoning.

Formally, given a high-FPS video v, a high-FPS selector retains a dense temporal stream and a tokenizer converts it into a sequence of visual tokens:

$$\tau(v) = \text{Tokenize}\big(\text{SelectHighFPS}(v)\big) \tag{1}$$

These visual tokens, together with the text tokens T representing the query, are then fed into a video LLM to generate the answer:

$$\hat{y} = \text{LLM}\big(\tau(v), T\big) \tag{2}$$

This formulation makes dense temporal information available to the model while leaving room for token-efficient representations.

3.2 Motion-Compensated Gated Inter-Tokenization

Existing video LLMs tokenize every frame and then prune or merge redundant tokens, resulting in unnecessary computation. To exploit the strong temporal redundancy in high-FPS videos, we draw inspiration from classical video compression techniques, which encode scenes using one full *key frame* and a series of *P-frames* that capture only the changing patches. Let $f_{s,k}$ denote the key frame of scene s, and let $f_{s,k+i}$ be the $(i+1)$-th frame. We define the P-frame residual as:

$$\Delta f_{s,k+j} = M_{s,k+j} \odot \big(f_{s,k+j} - f_{s,k+j-1}\big), \quad (3)$$

where $M_{s,k+j} \in \{0,1\}^{H\times W\times C}$ is a binary mask that selects only the pixel patches that have changed from the previous frame. Using these residuals, the frame sequence can be approximated from the key frame and its residual updates:

$$f_{s,k+i} = f_{s,k} + \sum_{j=1}^{i} \Delta f_{s,k+j}. \quad (4)$$

This compact representation preserves the visual changes most relevant to dense temporal perception while exposing only the dynamic patches, enabling tokenization complexity to grow sub-linearly with the number of frames.

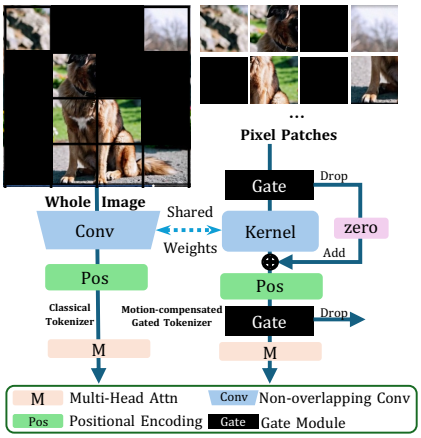

Fig. 3: Comparison of the classical tokenizer, such as CLIP [18] and SigLIP [27], and our Motion-Compensated Gated Inter-Tokenizer (MCG Tokenizer). **Left:** classical tokenizer, which applies a convolution over the entire image including masked patches, incurring high computational cost. **Right:** our MCG Tokenizer, which (1) reuses the pretrained convolutional kernels as an equivalent MLP, (2) applies a gating module to mask out irrelevant K- and P-patches (inserting zero-tensor placeholders for positional encoding), and (3) filters out placeholders before feeding only valid tokens into the multi-head attention. This parallelized pipeline dramatically reduces tokenization time.

Motion-Compensated Gated Tokenizer To efficiently tokenize only the dynamic patches identified by our Pixel-Coding Video Scene Representation, we repurpose the pretrained ViT tokenizer without additional training. We introduce a gating mechanism that filters out static patches using a binary mask derived from patch-wise structural similarity. Specifically, let $P_{s,j}^{(n)}$ denote the n-th patch of frame $f_{s,j}$ in scene s, and let $\mathrm{SSIM}(\cdot,\cdot)$ be the structural similarity index. We compute the mask entries $M_{s,j}^{(n)}$ as:

$$M_{s,j}^{(n)} = \begin{cases} 1 & \text{if } \mathrm{SSIM}\big(P_{s,j}^{(n)}, P_{s,j-1}^{(n)}\big) < \tau, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

where τ is a fixed threshold. We always set $M_{s,k}^{(n)} = 1$ for the key frame $f_{s,k}$. Next, the gating vector for frame $f_{s,j}$ is formed as

$$G_{s,j} = \left[M_{s,j}^{(1)}, M_{s,j}^{(2)}, \ldots, M_{s,j}^{(N)}\right], \tag{6}$$

where N is the total number of patches per frame. The gated tokenizer applies $G_{s,j}$ to the ViT embedding layer, processing only the selected patches and inserting zero-vectors for masked positions prior to positional encoding. This design reduces tokenization complexity to sub-linear in the number of frames while preserving all informative content.

Tokenizer Architecture The standard video tokenizer comprises a convolutional patch embedding layer, positional embeddings, and a 26-layer Transformer encoder. Because the embedding convolution uses non-overlapping kernels (stride = kernel size), we can flatten it into an equivalent MLP. Let p_n be the n-th patch in the gated sequence and $M_n \in \{0, 1\}$ its gate mask. We first compute the patch embedding:

$$e_n = \begin{cases} W_c\, p_n + b_c, & M_n = 1, \\ \mathbf{0}, & M_n = 0, \end{cases} \tag{7}$$

where W_c and b_c are the convolutional kernel weights and bias.

Next, we insert zero placeholders for masked positions and apply positional encoding:

$$\tilde{e}_n = \mathrm{PE}(e_n). \tag{8}$$

Finally, the sequence $\{\tilde{e}_n\}_{n=1}^{N}$ is processed by the Transformer encoder:

$$v = \mathrm{Transformer}\big(\tilde{e}_1, \tilde{e}_2, \ldots, \tilde{e}_N\big). \tag{9}$$

By gating patches before the embedding layer and using placeholders during positional encoding, we remove static patches while preserving token positions, achieving sub-linear growth in computation with respect to frame count.

3.3 Semantic-Coding Scene Merging

Pixel-level residual coding captures only low-level changes and lacks semantic context, which is insufficient for high-FPS video understanding. Incorporating deep semantic features often incurs significant computational overhead. However, pretrained vision-tokenizers already embed rich semantic information into discrete tokens. We therefore leverage these existing tokens to perform scene-level merging without retraining.

Let $\mathcal{T}_{s,k}$ be the set of tokens extracted from the key frame of scene s, and $\mathcal{T}_{s,k+j}$ the set of tokens from the j-th P frame. The full token sequence for scene s up to frame $k + i$ is:

$$\mathcal{T}_{s,k+i} = \mathrm{Concat}\left(\mathcal{T}_{s,k}, \mathcal{T}_{s,k+1}, \ldots, \mathcal{T}_{s,k+i}\right). \tag{10}$$

To merge semantically similar scenes, we compute a distance between their key-token distributions. For example, using cosine distance:

$$d\big(\mathcal{T}_{s,k}, \mathcal{T}_{t,k}\big) = 1 - \frac{\langle \mu(\mathcal{T}_{s,k}),\, \mu(\mathcal{T}_{t,k}) \rangle}{\|\mu(\mathcal{T}_{s,k})\|\, \|\mu(\mathcal{T}_{t,k})\|}, \tag{11}$$

where $\mu(\cdot)$ computes the mean embedding of a token set. If $d(\mathcal{T}_{s,k}, \mathcal{T}_{t,k}) < \delta$, we merge scene t into s by concatenating its P-frame tokens onto $\mathcal{T}_{s,k}$. This preserves dynamic information while eliminating redundant key-frame tokens.

By merging semantically similar scenes, we further compress the token sequence without losing critical static or dynamic content, enabling efficient high-FPS video processing.

Scene Merging After gated tokenization, each scene s is represented by a *key-token set* $\mathcal{T}_{s,k}$ (from the K-frame) and a sequence of *P-token sets* $\{\mathcal{T}_{s,k+1}, \ldots, \mathcal{T}_{s,k+i}\}$. Since key-token sets dominate the total token count, we reduce redundancy by merging semantically similar scenes based on the Jensen–Shannon divergence between their normalized token distributions:

$$D_{\mathrm{JSD}}(s,t) = \mathrm{JSD}\big(P(\mathcal{T}_{s,k}),\, P(\mathcal{T}_{t,k})\big), \tag{12}$$

where $P(\mathcal{T})$ denotes the frequency distribution of tokens in $\mathcal{T}$.

For each scene s, we identify the most semantically similar adjacent scene t by minimizing $D_{\mathrm{JSD}}(s,t)$, and merge the two scenes accordingly. The merged key-token set $\mathcal{T}'_{s,k}$ is formed by averaging the mean embeddings of both sets:

$$\mathcal{T}'_{s,k} = \frac{1}{2}\Big(\mu(\mathcal{T}_{s,k}) + \mu(\mathcal{T}_{t,k})\Big), \tag{13}$$

where $\mu(\mathcal{T})$ computes the mean token embedding of $\mathcal{T}$. The corresponding P-token sequences are then concatenated in temporal order:

$$\{\mathcal{T}_{s,k+1}, \ldots, \mathcal{T}_{s,k+i},\, \mathcal{T}_{t,k+1}, \ldots, \mathcal{T}_{t,k+j}\}. \tag{14}$$

This adaptive merging strategy removes redundant static content while preserving dynamic information, effectively reducing the overall token sequence length.

3.4 Integration Details

After token-scene merging, we flatten the token sets into a single sequence and pass them through a linear vision-projection layer. These visual tokens are then concatenated with the query prompt and any additional text tokens before being fed, without further training, into the video LLM to generate the answer. This architecture enhances the model's ability to process dense temporal information while remaining compatible with standard LLM pipelines.

4 DIVE-Bench

4.1 Benchmark Construction

Although we define the High-FPS Video Understanding task and propose a corresponding model, evaluating this capability requires a benchmark that preserves dense temporal information rather than collapsing videos through sparse sampling. Existing benchmarks largely rely on low-FPS QA and therefore fail to assess frame-by-frame reasoning.

We introduce **DIVE-Bench** (Dense Information Video Evaluation Benchmark), a benchmark designed to stress-test both temporal reasoning and token throughput of video LLMs under high-FPS conditions. Instead of collecting new data, DIVE-Bench reuses existing video datasets together with their dense annotation streams, thereby preserving fine-grained temporal fidelity. As the first benchmark explicitly targeting high-FPS video understanding, DIVE-Bench focuses on videos in which informative content appears in nearly every frame, enabling the construction of meaningful dense QA pairs. We consider two complementary scenarios to pioneer high-FPS video QA: educational videos with rapidly changing textual content and high-motion egocentric videos with fast object movements. **Additional details are provided in the supplementary materials**.

4.2 Educational High-FPS Videos

We source long-form lecture videos from YouTube (e.g., LPMDataset [7]) with durations exceeding 30 minutes. At standard frame rates, each clip contains over 10^5 frames, forcing models to maintain a high effective FPS to capture all relevant information.

Dense QA construction. As illustrated in Fig. 1, we design QA pairs that explicitly require dense temporal processing by treating embedded subtitles as ground-truth answers. Subtitle lines typically change at sub-second intervals; reconstructing the full subtitle stream therefore requires inspecting thousands of frames. Models that sample too sparsely will inevitably hallucinate or omit subtitle segments, leading to systematic errors. This design allows DIVE-Bench to explicitly *track information loss* caused by insufficient temporal sampling, ensuring that strong performance requires both high-FPS processing and the ability to handle long-duration video without losing frame-level information.

4.3 High-Motion High-FPS Videos

To complement dense textual understanding, we construct a high-motion scenario using egocentric datasets with rapidly moving objects (e.g., hands in EgoDex [5]). In these clips, salient visual information changes almost every frame, making sparse frame sampling ineffective.

To enable a training-free evaluation compatible with language-based VLLMs, we avoid direct coordinate regression. Instead, we overlay a 3×3 grid on each

frame and ask the model to predict the grid cell containing the palm center (e.g., `r1c2`). The predicted grid index is mapped to a coarse 2D location using the cell center, converting dense localization into a discrete classification problem while still requiring frame-by-frame inspection.

Dense QA construction. Using ground-truth palm-center annotations, we derive the correct grid label for each frame and treat every frame-label pair as a QA instance. Accuracy thus directly reflects a model's ability to process densely sampled frames. Together with the educational-video setting, this high-motion scenario covers two representative forms of dense temporal information: rapidly changing text and rapidly changing motion.

5 Experiments

5.1 Implementation Details

All experiments are performed in a zero–shot, inference-only setting using both 0.5B- and 7B-parameter variants of our model, built on the LLaVA-One Vision framework [10] and the QWen2 architecture [21]. We tokenize 224x224 frames into 16x16 patches using `QwenTokenizerFast`. Inference runs on NVIDIA RTX 4090, RTX A6000 Ada, and H200 GPUs with mixed-precision (FP16). We measure *tokenization time* from raw frame extraction through completion of the token sequence, averaged over 50 videos using the LMMS-Eval toolkit [35]. Subjective quality is assessed via Mean Opinion Scores (MOS) assigned by GPT-3.5 [17], where each generated answer is rated against the ground truth on a 0–5 scale.

5.2 Comparative Methods

Since our method operates directly on visual tokenization and requires access to the input token pipeline, we restrict our comparisons to open-source video large language models and do not consider proprietary or closed-source systems. As our work is the first to explicitly target the task of high-FPS video understanding, there is no existing method directly comparable to ours. To enable a fair and controlled evaluation, we adapt several recent video large language models (video-LLMs) as baselines by modifying their backbones to support denser frame processing. We compare our approach against five recent video-LLM systems. 1) LLaVA-One Vision [8] samples a fixed number of frames at uniform intervals before concatenating visual and text tokens. Its flexible FPS variant [6] dynamically adjusts the sampling rate to focus on high-motion segments. 2) LLaVA-Video [36] integrates temporal-attention modules into the LLaVA backbone to capture inter-frame motion cues. 3) LLaVA-Next SI [10] introduces spatial-instruction tuning and temporal-aware modifications into LLaVA with single image finetuning to improve instructional video understanding.

5.3 Metrics

As DIVE-Bench is the first benchmark targeting high-FPS video understanding, we evaluate models using task-specific metrics tailored to different video characteristics. For educational and lecture-style high-FPS videos, we adopt the *Mean Opinion Score (MOS)* [20] to assess answer quality. MOS measures perceived correctness and completeness on a 0–5 scale to ensure consistent and scalable evaluation. We also report *Character Error Rate (CER)*, *Word Error Rate (WER)*, *Token-level F1 (Token-F1)*, and *Exact Match (EM)* to quantify transcription-style accuracy at different granularities. In addition, we report *Effective FPS*, defined as the achieved processing frame rate during inference, and *Post Tokens*, the number of visual tokens retained after token reduction, which together characterize high-FPS scalability and token efficiency. We further report *Tokenization Time*, defined as the wall-clock time required to convert raw video frames into visual tokens, which directly reflects preprocessing efficiency under high-FPS inputs. For high-motion high-FPS videos, where fine-grained temporal dynamics are critical and long-form textual answers are less suitable, we employ *Grid Average Displacement Error (Grid ADE)* and *Grid Final Displacement Error (Grid FDE)*. These metrics provide a fair and stable comparison for motion-sensitive scenarios by measuring spatial prediction accuracy over time. Detailed definitions and computation procedures for all metrics are provided in the supplementary material.

5.4 Quantitative Results

Educational High-FPS Videos Table 1 presents the Mean Opinion Scores (MOS) on DIVE-Bench. Our 0.5B-parameter GRT model achieves the strongest MOS among the listed open-source baselines, including larger 7B-parameter models. These results suggest that codec-style gated tokenization can improve answer quality on dense temporal perception tasks without increasing model size.

Table 1: MOS on DIVE-Bench. Our 0.5B GRT backbone achieves the strongest MOS among the listed open-source baselines.

Method	#Params	MOS
LLaVA-Video [36]	7.0B	1.47
LLaVA-OneVision [8]	7.0B	1.70
LLaVA-OneVision [8]	0.5B	2.01
LLaVA-Next SI [10]	0.5B	1.73
GRT (Ours)	0.5B	**2.50**

Table 2: Subtask performance on MLVU under matched low-FPS sampling.

Category	Baseline	Ours (GRT)
TutorialQA	27.900	27.900
Ego-centric	37.700	39.600
Anomaly Recognition	30.800	38.500
Topic Reasoning	62.600	62.600
Average	33.000	34.100

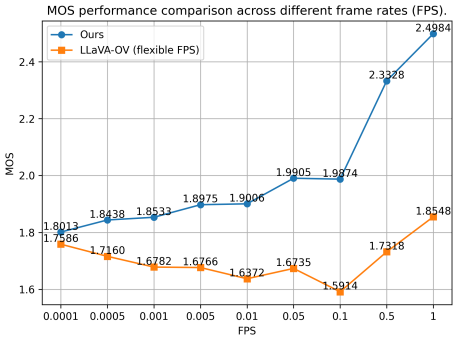

Fig. 4: MOS versus input frame rate (FPS) for GRT and the modified LLaVA-OV baseline. Denser sampling provides more temporal evidence for subtitle recovery, and GRT maintains stronger quality across the tested range.

Gated Tokenizer	Scene Merge	MOS
—	—	1.66
✓	—	1.93
✓	✓	**1.94**

Table 4: Ablation results showing the impact of the gated tokenizer and scene-merge modules. "—" indicates disabled, and "✓" indicates enabled.

Reduction Stage	0.01 FPS	0.1 FPS	1 FPS
Gated Pruning	1.00	0.96	0.90
Scene Merging	1.00	0.33	0.14

Table 5: Effect of frame rate on token retention after each reduction stage. Values represent the proportion of tokens retained relative to the baseline.

Table 3: Dense motion understanding on high-motion videos (grid-based evaluation). We report grid ADE/FDE on the 3×3 palm-location prediction task. Our method achieves lower grid ADE/FDE while operating at a higher effective FPS.

Method	Effective FPS ↑	Grid ADE ↓	Grid FDE ↓
Baseline (LLaVA-OV)	4.49	1.3618	1.6085
Ours (GRT)	8.69	**1.1456**	**1.1828**

High-Motion High-FPS Video Across the high-motion benchmark, our GRT-based configuration achieves both higher effective FPS and lower grid errors compared to the LLaVA-OV baseline (Table 3). In particular, increasing the effective FPS from 4.49 to 8.69 is accompanied by reductions in Grid ADE and Grid FDE, demonstrating that efficient tokenization allows the model to benefit from denser frame sampling rather than being bottlenecked by token budget. This supports our main claim that dense temporal modeling and token-efficiency must be co-designed for high-motion video understanding.

5.5 Impact of Frame Rate

To assess robustness to varying temporal resolutions, we vary the input frame rate from 0.0001 FPS to 1.0 FPS. For a controlled comparison, we remove the maximum-frame constraint from the LLaVA-OV baseline, ensuring both methods share the same model size (0.5B). Figure 4 plots MOS as a function of FPS. Denser inputs generally provide more recoverable temporal evidence for subtitle reconstruction, while extremely low-FPS inputs lead to severe omissions. Across the tested range, GRT maintains stronger answer quality than the modified baseline, underscoring the need for high-FPS evaluation in VLLMs.

5.6 Ablation Study

To quantify the contributions of our two main components, the Motion-Compensated Gated Inter-Tokenizer ("Gated Tokenizer") and the Semantic-Scene Merging module ("Scene Merge"), we perform an ablation study on DIVE-Bench. Table 4 reports both accuracy and MOS for three configurations: **Baseline (no components)**: raw full-FPS tokenization without any gating or merging. **Gated Tokenizer only**: applies pixel-coding gating to prune static patches before tokenization. **Full model**: combines the Gated Tokenizer with Semantic-Scene Merging. Enabling the Gated Tokenizer alone yields a significant boost in both accuracy and MOS compared to the baseline, demonstrating the effectiveness of early patch pruning. Adding Scene Merge provides additional token compression and a small MOS gain in this setting Overall, gated inter-tokenization is the primary quality-preserving component, while scene merging is best viewed as an optional compression stage for improving the efficiency-quality trade-off.

5.7 Further Evaluation on Existing Benchmarks

To demonstrate that our approach generalizes beyond DIVE-Bench, we further evaluate GRT on two widely used long-form video understanding benchmarks: MLVU [37] and VideoMME [4]. We apply our method under the same frame-sampling configuration as the LLaVA-One Vision (0.5B) baseline [10] to ensure a controlled and fair comparison.

As shown in Table 7, GRT is compatible with both benchmarks under the matched low-FPS setting, with a clearer gain on MLVU and a nearly neutral result on VideoMME. These results indicate that our method can be layered onto conventional long-video evaluation settings, although its main advantage is expected in dense temporal perception regimes where higher effective FPS is necessary.

5.8 Semantic Fidelity and Motion Preservation Analysis

Gated Residual Tokenization (GRT) reduces computational cost by pruning redundant visual tokens before and after tokenization. While this strategy significantly improves efficiency, it is important to examine whether aggressive token gating may compromise semantic fidelity, particularly for subtle motion patterns or under challenging conditions such as camera shake or partial occlusion.

At present, there is no established benchmark with ground-truth annotations that directly quantify missed-motion recall or semantic consistency under dense temporal distortions. Such evaluations would require fine-grained, pixel-level motion annotations across long video sequences, which are not available in existing dense-temporal datasets. As a practical proxy, we analyze performance across fine-grained temporal reasoning categories in MLVU [37], which includes diverse motion-sensitive subtasks.

As shown in Table 2, GRT achieves notable improvements on motion-sensitive categories such as *ego-centric understanding* and *anomaly recognition*, indicating that fine-grained motion cues are effectively preserved despite token pruning.

Meanwhile, performance remains stable on other subtasks, suggesting that semantic correctness is not degraded under complex temporal conditions. These results provide empirical evidence that GRT maintains semantic fidelity while improving efficiency, even without explicit supervision for missed-motion detection.

5.9 Tokenization Time and Token Reduction under Different FPS

We evaluate both tokenization latency and token reduction effectiveness at three representative frame rates: 0.01, 0.1, and 1 FPS. Table 6 reports the average time to tokenize a short video segment for our method versus the LLaVA-OV baseline (which processes every patch without gating or merging). Timing is measured by instrumenting the tokenizer entry and exit points on an NVIDIA RTX 4090. At 1 FPS, our motion-compensated gated tokenizer reduces latency by 53.6%, from 0.0487 s to 0.0226 s. At lower frame rates, latency is comparable: GRT is slightly slower at 0.01 FPS and 4.8% faster at 0.1 FPS, indicating that early patch pruning becomes most beneficial as the number of frames grows.

Table 6: Tokenization time at different input sampling rates (FPS). We report per-video tokenization latency (seconds) for LLaVA-OV and GRT under matched model size and sampling settings. Notably, Gemini's official video understanding pipeline (File API / Vertex AI) samples uploaded videos at 1 FPS by default, which makes efficient processing at $\geq$1 FPS particularly relevant for high-FPS video understanding. *Speedup* is computed as $(t_{\text{base}} - t_{\text{ours}})/t_{\text{base}}$.

Method	0.01 FPS	0.1 FPS	1 FPS
LLaVA-OV [8]	0.0170 s	0.0186 s	0.0487 s
GRT (Ours)	0.0174 s	0.0177 s	0.0226 s
Speedup	-2.4%	4.8%	53.6%

Table 7: Performance comparison on long-video benchmarks **MLVU** [37] and **VideoMME** [4] under the same frame-sampling configuration. Since these benchmarks primarily evaluate long-horizon semantic understanding with sparse sampling rather than dense temporal reasoning, we report results under a shared low-FPS setting to ensure a fair, controlled comparison with the baseline.

Method	MLVU	VideoMME
LLaVA-One Vision (0.5B)	33.002	43.963
Ours (GRT, 0.5B)	34.066	44.037
Δ (Ours − Base)	+1.064	+0.074

To quantify token savings, Table 5 shows the fraction of original tokens retained after each reduction stage. At 0.01 FPS, both gated pruning and scene merging have little effect, since few inter-frame changes occur. At 0.1 FPS, gated pruning alone retains 96% of tokens, while combining with scene merging reduces this to 33%. At 1 FPS, gated pruning retains 90% of tokens, and scene merging further reduces the sequence to just 14%. These results confirm that our two-stage reduction, first at the patch level, then at the scene level, becomes increasingly effective as FPS increases.

5.10 Additional DIVE-Bench diagnostics.

The additional diagnostics support two camera-ready claims. First, the released data audit finds no missing paths, empty answers, duplicate video identifiers, or detected temporal leakage in the validated partitions; the high-motion partition keeps invalid grid parses and repeated question patterns as explicit audit flags. Second, the model-family check shows that Qwen2.5-VL gives the strongest high-motion results, OneVision-Encoder gives the highest educational token-overlap score but weaker mean opinion score, and adding our codec-style method improves LLaVA-One Vision and Qwen2.5-VL on most dense-motion metrics. We omit the high-frame-rate diagnostic table from the main paper because it is mainly a stress test of effective frame-rate saturation rather than a primary accuracy result; across 1–30 requested frames per second, character error stays within 1.002–1.021, word error stays within 1.007–1.023, and token-overlap score stays within 0.004–0.006.

Table 8: Dataset statistics.

Partition	Videos or clips	Question-answer pairs	Dense signal
Educational dense videos	317 videos	317	Subtitle sequence
High-motion dense videos	277 clips	3243	Grid trajectory sequence

Table 9: DIVE-Bench differs from conventional long-video benchmarks in output granularity.

Benchmark	Answer format	Dense frame-aligned supervision
VideoMME	Multiple-choice question answering	No
LongVideoBench	Multiple-choice question answering with referred context	No
DIVE-Bench educational partition	Subtitle sequence	Yes
DIVE-Bench high-motion partition	Grid trajectory sequence	Yes

The educational partition averages 44.26 source frames per second and 72068.5 frames per video, while the high-motion partition averages 30.00 source frames per second and 254.8 frames per clip. The audit has clean path and split statistics: educational dense videos contain no missing paths, empty answers, duplicate question/video pairs, or temporal overlaps, with only 6 repeated subtitle answers (1.89%); high-motion dense videos contain no missing paths, empty answers, duplicate clip identifiers, or detected split leakage, while 244 invalid or empty grid parses (7.52%) and 171 repeated question patterns (5.27%) are retained only as released audit flags.

These additional runs suggest that the benchmark is not tied to a single model family. LLaVA-One Vision benefits from our method on mean opinion score and high-motion metrics, Qwen2.5-VL with our method gives the strongest

high-motion numbers, and OneVision-Encoder is useful as a complementary text-overlap reference but has lower subjective answer quality. Since these cross-family results are diagnostic and the gains are small, we report them as supporting evidence rather than as an additional main table.

6 Limitations and Future Work

DIVE-Bench is an initial step toward codec-style high-frame-rate video understanding. Its current question-answer construction focuses on two dense signals: caption or optical-character-recognition streams in educational videos and grid-based motion trajectories in high-motion videos. This makes the benchmark auditable and frame-aligned, but it does not cover every form of dense temporal perception. Future versions can extend the same principle to real-time video streams, motion-centric game videos, and medical or surgical videos, where missing a small temporal transition can change the semantic interpretation. Our current method also accelerates visual tokenization under an existing sampled frame rate. A next step is to use priors from real video codecs, such as key frames, motion vectors, and residual streams, so that video-language models can encode high-frame-rate input without relying on skipped frames.

7 Conclusion

We introduced the task of **High-FPS Video Understanding**, which requires video LLMs to reason over densely sampled, high-FPS content. To support this, we proposed **DIVE-Bench**, the first benchmark for evaluating fine-grained temporal understanding in video QA. To address the inefficiency of high-frame-rate tokenization, we developed *Gated Residual Tokenization (GRT)*, a codec-style framework that combines motion-compensated gated inter-tokenization and semantic-scene token merging to reduce token count with minimal information loss. Experiments show that GRT improves the efficiency-quality trade-off on DIVE-Bench, while accelerating tokenization by up to 53.6%. Ablation studies identify gated inter-tokenization as the primary quality-preserving component and semantic-scene merging as a complementary compression stage, highlighting the value of dense temporal modeling for video large language models.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems **35**, 23716–23736 (2022)
2. Fei, Y., Gao, Y., Xian, X., Zhang, X., Wu, T., Chen, W.: Do current video llms have strong ocr abilities? a preliminary study. In: Proceedings of the 31st International Conference on Computational Linguistics. pp. 9860–9876 (2025)
3. Felsen, P., Lucey, P., Ganguly, S.: Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders. In: Proceedings of the European conference on computer vision (ECCV). pp. 732–747 (2018)

4. Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., et al.: Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 24108–24118 (2025)
5. Hoque, R., Huang, P., Yoon, D.J., Sivapurapu, M., Zhang, J.: Egodex: Learning dexterous manipulation from large-scale egocentric video (2025), https://arxiv.org/abs/2505.11709
6. Kim, W., Choi, C., Lee, W., Rhee, W.: An image grid can be worth a video: Zero-shot video question answering using a vlm. IEEE Access (2024)
7. Lee, D.W., Ahuja, C., Liang, P.P., Natu, S., Morency, L.P.: Lecture presentations multimodal dataset: Towards understanding multimodality in educational videos. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 20087–20098 (2023)
8. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al.: Llava-onevision: Easy visual task transfer. Transactions on Machine Learning Research (2025)
9. Li, B., Shen, Y., Liu, Y., Xu, Y., Liu, J., Li, X., Li, Z., Zhu, J., Zhong, Y., Lan, F., et al.: Toward cognitive supersensing in multimodal large language model. arXiv preprint arXiv:2602.01541 (2026)
10. Li, F., Zhang, R., Zhang, H., Zhang, Y., Li, B., Li, W., Ma, Z., Li, C.: Llava-interleave: Tackling multi-image, video, and 3d in large multimodal models. In: International Conference on Learning Representations. vol. 2025, pp. 81182–81199 (2025)
11. Litzenberger, M., Belbachir, A.N., Schon, P., Posch, C.: Embedded smart camera for high speed vision. In: 2007 First ACM/IEEE International Conference on Distributed Smart Cameras. pp. 81–86. IEEE (2007)
12. Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., Lee, Y.J.: Llava-next: Improved reasoning, ocr, and world knowledge (January 2024), https://llava-vl.github.io/blog/2024-01-30-llava-next/
13. Lu, J., Wang, H., Ma, X., Dong, Q., Zhang, M., Wang, Y., Fu, Y.: Muse: A unified agentic harness for mllms. arXiv preprint arXiv:2606.03005 (2026), https://arxiv.org/abs/2606.03005
14. Lu, J., Wang, H., Xu, Y., Wang, Y., Yang, K., Fu, Y.: Representation potentials of foundation models for multimodal alignment: A survey. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 16669–16684. Association for Computational Linguistics (Nov 2025)
15. Lu, J., Wang, H., Yang, K., Zhang, Y., Jenni, S., Fu, Y.: The indra representation hypothesis for multimodal alignment. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2026), https://openreview.net/forum?id=D2NR5Zq6PG
16. Maaz, M., Rasheed, H., Khan, S., Khan, F.: Video-chatgpt: Towards detailed video understanding via large vision and language models. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 12585–12602 (2024)
17. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
18. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from

natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
19. Shen, Y., Liu, J., Li, X., Liu, Y., Li, B., Yang, H., Jia, W., Li, Y., Yu, T., Rehg, J.M., et al.: Egoforge: Goal-directed egocentric world simulator. arXiv preprint arXiv:2603.20169 (2026)
20. Streijl, R.C., Winkler, S., Hands, D.S.: Mean opinion score (mos) revisited: methods and applications, limitations and alternatives. Multimedia Systems **22**(2), 213–227 (2016)
21. Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., et al.: Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191 (2024)
22. Wang, T., Li, Z., Lin, M., Luo, S., Shen, Y., Bera, A., Yang, B., Chen, Y.V.: Finemola: Towards fine-grained motion-language alignment from clip-level supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5510–5519 (2026)
23. Wang, Y., He, Y., Li, Y., Li, K., Yu, J., Ma, X., Li, X., Chen, G., Chen, X., Wang, Y., et al.: Internvid: A large-scale video-text dataset for multimodal understanding and generation. In: The Twelfth International Conference on Learning Representations (2023)
24. Xu, Z., Wang, Y., Abbatematteo, B., Preechayasomboon, J., Chan, S., Colonnese, N., Memar, A.H.: Contact-grounded policy: Dexterous visuotactile policy with generative contact grounding (2026), https://arxiv.org/abs/2603.05687
25. Xue, H., Hang, T., Zeng, Y., Sun, Y., Liu, B., Yang, H., Fu, J., Guo, B.: Advancing high-resolution video-language representation with large-scale video transcriptions. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
26. Zellers, R., Lu, J., Lu, X., Yu, Y., Zhao, Y., Salehi, M., Kusupati, A., Hessel, J., Farhadi, A., Choi, Y.: Merlot reserve: Neural script knowledge through vision and language and sound. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022. pp. 16354–16366. IEEE Computer Society (2022)
27. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11975–11986 (2023)
28. Zhang, H., Chen, M., He, S., Xu, Z., Shen, Y., Huang, Y., Lu, J., Li, Y., She, Y., Fu, Y.: A survey of physical ai: A history from chatgpt to world models and embodied agents. Preprints (June 2026). https://doi.org/10.20944/preprints202606.0173.v1, https://doi.org/10.20944/preprints202606.0173.v1
29. Zhang, H., Fu, Y.: Vqtoken: Neural discrete token representation learning for extreme token reduction in video large language models. Advances in Neural Information Processing Systems **38**, 32851–32869 (2026)
30. Zhang, H., Li, Y., He, S., Nagarajan, T., Chen, M., Lu, J., Li, A., Fu, Y.: Thinkjepa: Empowering latent world models with large vision-language reasoning model. arXiv preprint arXiv:2603.22281 (2026)
31. Zhang, H., Xu, Y., Fu, Y.: Out-of-sight embodied agents: Multimodal tracking, sensor fusion, and trajectory forecasting. IEEE Transactions on Pattern Analysis and Machine Intelligence (2026)
32. Zhang, H., Xu, Y., Lu, H., Shimizu, T., Fu, Y.: Layout sequence prediction from noisy mobile modality. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 3965–3974 (2023)

33. Zhang, H., Xu, Y., Lu, H., Shimizu, T., Fu, Y.: Oostraj: Out-of-sight trajectory prediction with vision-positioning denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14802–14811 (2024)
34. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding. arXiv preprint arXiv:2306.02858 (2023), https://arxiv.org/abs/2306.02858
35. Zhang, K., Li, B., Zhang, P., Pu, F., Cahyono, J.A., Hu, K., Liu, S., Zhang, Y., Yang, J., Li, C., et al.: Lmms-eval: Reality check on the evaluation of large multimodal models. arXiv preprint arXiv:2407.12772 (2024)
36. Zhang, Y., Wu, J., Li, W., Li, B., Ma, Z., Liu, Z., Li, C.: Video instruction tuning with synthetic data (2024), https://arxiv.org/abs/2410.02713
37. Zhou, J., Shu, Y., Zhao, B., Wu, B., Liang, Z., Xiao, S., Qin, M., Yang, X., Xiong, Y., Zhang, B., et al.: Mlvu: Benchmarking multi-task long video understanding. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 13691–13701 (2025)
38. Zhu, K., Zhao, Y., Zhao, L., Zuo, G., Gu, Y., Xie, D., Gao, Y., Xu, Q., Tang, T., Ye, Z., et al.: Nanoflow: Towards optimal large language model serving throughput. arXiv preprint arXiv:2408.12757 (2024)