ReMoRa: Multimodal Large Language Model based on Refined Motion Representation for Long-Video Understanding

Supplementary Material

A. Deep State Space Models

State space models (SSMs) [11, 13, 14] have demonstrated strong capability in modeling long-range temporal dependencies while maintaining computational efficiency. SSMs are inspired by control theory [20], in which a temporal process $\boldsymbol{x}(t) \in \mathbb{R} \mapsto \boldsymbol{y}(t) \in \mathbb{R}$ is represented by a Q-dimensional hidden state $\boldsymbol{h}(t) \in \mathbb{R}^Q$ as follows:

$$\frac{d\boldsymbol{h}(t)}{dt} = \mathbf{A}\boldsymbol{h}(t) + \mathbf{B}\boldsymbol{x}(t), \tag{8}$$

$$y(t) = \mathbf{C}\boldsymbol{h}(t) + D\boldsymbol{x}(t), \tag{9}$$

where $\mathbf{A} \in \mathbb{R}^{Q\times Q}$ governs the latent dynamics, and $\mathbf{B}, \mathbf{C}, D$ are projection matrices. By discretizing these continuous dynamics with a timescale parameter Δ and applying the zero-order hold [73], we obtain:

$$\boldsymbol{h}_j = \bar{\mathbf{A}}\boldsymbol{h}_{j-1} + \bar{\mathbf{B}}\boldsymbol{x}_j, \tag{10}$$

$$y_j = \mathbf{C}\boldsymbol{h}_j + D\boldsymbol{x}_j. \tag{11}$$

Here, $\bar{\mathbf{A}} = \exp(\Delta\mathbf{A})$ and $\bar{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B}$. While the recursive update resembles RNNs, its sequential nature hinders parallelization. To address this, S4 [14] reformulates the system into a convolutional form:

$$\bar{\mathbf{K}} = \left(\mathbf{C}\bar{\mathbf{B}} + D,\ \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}} + D, \ldots, \mathbf{C}\bar{\mathbf{A}}^{L-1}\bar{\mathbf{B}} + D\right), \tag{12}$$

$$\boldsymbol{y} = \bar{\mathbf{K}} * \boldsymbol{x}, \tag{13}$$

where L denotes sequence length. This formulation enables parallel training via convolution (Eq. (13)) and efficient inference through recurrence (Eq. (10), (11)).

Building on this, Mamba [13] introduces a dynamic selection mechanism where $\bar{\mathbf{A}}, \bar{\mathbf{B}}$, and $\bar{\mathbf{C}}$ are conditioned on the input $\boldsymbol{x}$, allowing time-varying transitions that enhance expressive capacity. Consequently, Mamba achieves superior performance on long-sequence modeling tasks, outperforming transformers in several language and temporal domains. Crucially, SSMs scale linearly with sequence length, offering a significant computational advantage over the quadratic complexity of standard transformers for long sequences.

Beyond generic sequence benchmarks, deep SSMs (e.g., S4, S5, and Mamba [13, 14, 50]) have been adopted for long-sequence modeling across a wide range of domains, including robotics and video understanding [28, 34, 40]. In these settings, SSMs act as temporal backbones for models that must process very long sequences, often matching or surpassing transformer-based architectures at lower computational cost [11, 13, 14]. However, existing video-focused SSMs [7, 28] primarily operate on dense RGB frame sequences and have not been adapted to exploit compressed-domain motion cues in multimodal video-language settings as considered in this work.

Model	MotionBench							LVB
	Ave.	MR	LM	CM	MO	AO	RC	
Qwen2-VL-7B	0.52	0.52	0.55	**0.49**	0.68	0.39	0.32	55.6
Qwen3-VL-8B	0.53	0.55	0.56	0.48	0.68	0.39	0.24	58.0
ReMoRa (Ours)	**0.55**	**0.59**	**0.57**	**0.49**	**0.70**	**0.40**	**0.37**	**60.8**

Table 8. Results on MotionBench and LongVideoBench (LVB). The Average (Ave.) score applies to MotionBench.

B. Additional Results

Motion-specific Evaluation. To directly evaluate the benefit of our refined motion representation, we conducted experiments on MotionBench [15], a benchmark designed to probe motion understanding through fine-grained subcategories: Motion Recognition (MR), Location of Motion (LM), Camera Motion (CM), Motion Order (MO), Attribute of Object (AO), and Rotation Count (RC). As shown in Table 8, ReMoRa achieved the best average score of 0.55, outperforming both Qwen2-VL-7B (0.52) and the recent Qwen3-VL-8B (0.53), despite the latter employing a stronger backbone and training recipe. ReMoRa also outperformed both baselines on most subcategories, with notable gains on Motion Recognition and Rotation Count, confirming that its performance gains stem from enhanced motion representation.

Preprocessing Overhead. Table 9 reports the per-video wall-clock time for different frame sampling strategies, measured on 100 videos from LongVideoBench (3 runs per video, averaged). Our scene-adaptive pipeline (64 I-frames + up to 32 P/B-frames per I-frame) takes 3.12 s per video, only 1.07 s more than uniform 64-frame RGB sampling (2.05 s). To match our temporal coverage with dense RGB decoding ($64 + 64 \times 32 = 2{,}112$ frames), naive RGB extraction would require 58.51 s per video, making it impractical at scale. These results show that compressed-domain P/B-frames provide denser temporal cues at practical latency.

C. Benchmarks

LongVideoBench. LongVideoBench [57] is a long-context video question answering benchmark constructed

Setting	Time (s)
Uniform 32	1.03
Uniform 64	2.05
Uniform 128	4.03
1 fps	12.49
Naive RGB	58.51
Ours	3.12

Table 9. Per-video wall-clock time on 100 videos from LongVideoBench.

from 3,763 web videos and 6,678 human-written multiple-choice questions. It considers contexts of up to one hour and includes aligned textual signals such as subtitles, with a design that explicitly targets long-range temporal reasoning rather than short clip understanding.

MLVU. MLVU [76] targets comprehensive long-video comprehension, spanning videos from a few minutes to nearly two hours. The videos are drawn from diverse sources, including movies, surveillance footage, egocentric recordings, and gameplay. The benchmark evaluates multiple tasks, such as open-ended question answering, multiple-choice question answering, and temporal localization, providing a broad view of long-form video understanding.

NExT-QA. NExT-QA [58] contains 5,440 videos and approximately 52,000 manually annotated question–answer pairs. It is specifically designed to probe causal, temporal, and intentional reasoning about human activities, using both multiple-choice and open-ended questions to assess higher-level understanding beyond surface description.

VideoMME. VideoMME [12] is a large-scale evaluation suite for video multimodal large language models. It comprises roughly 900 videos totaling about 254 hours, with approximately 2,700 human-authored question–answer pairs. The benchmark covers short, medium, and long videos and supports multiple modalities, including visual frames, subtitles, and audio, enabling a comprehensive assessment of multimodal reasoning.

Perception Test. Perception Test [44] is a diagnostic benchmark that measures core perceptual and reasoning skills such as memory, abstraction, physical reasoning, and semantic understanding. It uses real-world videos with dense human annotation across video, audio, and text, and is designed to evaluate generalization capabilities rather than simple pattern matching.

MSVD-QA. MSVD-QA [59] is derived from the MSVD captioning dataset by automatically converting captions for each short clip into question–answer pairs. Models are evaluated using LLM-as-a-judge, making it a standard benchmark for short video question answering.

Optimizer	AdamW
Learning rate	2e-5
Batch size	32
T	64
H, W	$384, 384$
K	64
N_m	32
p	16
b_h, b_w	$4, 4$
f_v	16 fps

Table 10. Hyperparameters used in our experiments.

ActivityNet-QA. ActivityNet-QA [65] consists of around 58,000 human-written question–answer pairs over roughly 5,800 videos. The benchmark focuses on reasoning over complex and temporally extended web videos, requiring models to integrate information across long and diverse activities.

MotionBench. MotionBench [15] is a benchmark specifically designed to evaluate motion understanding in video MLLMs. It contains fine-grained subcategories including Motion Recognition, Location of Motion, Camera Motion, Motion Order, Attribute of Object, and Rotation Count, enabling targeted diagnosis of a model's ability to perceive and reason about different aspects of motion in videos.

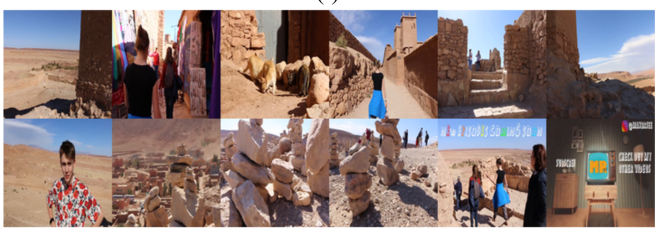
D. Implementation Details.

Our model comprised approximately 7.6B parameters and required around 2,900 T multiply–add operations. Training was conducted on 16 NVIDIA H200 SXM GPUs (141 GB VRAM), while evaluation was performed on a single H200 GPU. The total training time was approximately 21 hours. Table 10 summarizes the hyperparameters used in our main experiments.

E. Additional Qualitative Results

Fig. 4 presents additional qualitative examples from LongVideoBench. Panels (i) and (ii) each show sampled frames from the video, the question with multiple-choice options, and the predictions of ReMoRa and the baseline model. In example Fig 4 (i), the model must track how the appearance of the backpack changes as the woman moves through the scene and descends the hill. ReMoRa correctly reasons about the final state of the scene and selects the option de-

(i)



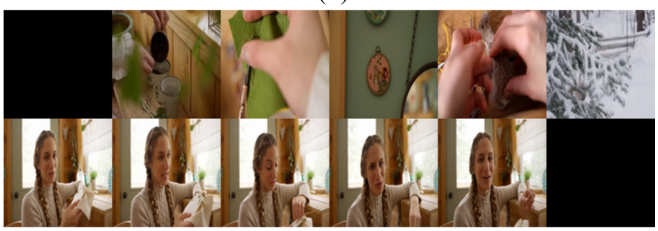
Question: At the beginning of the video, a woman with a headband tied to her head, wearing a red top, carrying a black backpack, when the woman comes down from a hill with tall rocks, what changes occur to her backpack?

Options:

[A] There is a dark red jacket hanging on her black backpack
[B] Nothing changed
[C] There is a white jacket hanging on her black backpack
[D] There is a dark blue jacket hanging on her black backpack

Ours: [D] **Baseline:** [A]

(ii)



Question: In a room, in front of a wooden table, there is a person wearing a light gray knitted sweater holding embroidery and a needle, doing embroidery. Who is this person doing the embroidery?

Options:

[A] The woman with loose brown hair
[B] The woman with loose black hair
[C] The woman with black braided hair
[D] The woman with brown braided hair

Ours: [D] **Baseline:** [C]

Figure 4. Further qualitative comparison between ReMoRa and LLaVA-Video on LongVideoBench. In both examples, ReMoRa correctly answers questions that require integrating spatial details with long-range temporal understanding, such as tracking how the scene and objects change over time and consistently identifying the person involved in the activity, while the baseline model fails.

Error mode	# instances
Spatial comprehension error	22
Temporal comprehension error	21
Motion comprehension error	14
Annotation error	10
Total	**67**

Table 11. Categorization of failure modes based on 67 annotated error instances from 50 randomly sampled cases where our model failed and LLaVA-Video succeeded on the NExT-QA benchmark. A single case may belong to multiple error modes.

scribing the dark blue jacket on the backpack, whereas the baseline prediction is inconsistent with the visual evidence.

Example Fig 4 (ii) requires consistent identification of the person doing embroidery across multiple shots with similar backgrounds and distracting context. ReMoRa correctly associates the description in the question with the woman who appears throughout the sequence and selects the option matching her hairstyle and appearance, while the baseline focuses on an incorrect description.

F. Error Analysis

To better understand the limitations of ReMoRa, we conducted an error analysis. We defined a failure case as a sample for which our model generated an incorrect answer while the baseline model (LLaVA-Video) generated the correct one. Out of 8,564 evaluation samples, this criterion yielded 270 failures for our model. From these, we randomly sampled 50 cases and manually analyzed their underlying causes. Each case could be assigned to multiple error categories, so the total count across all error types exceeds 50, resulting in 67 annotated error instances in total. Table 11 shows the distribution over four major error modes.

Spatial comprehension errors. This category covers failures related to spatial relationships, object localization, and object presence. Examples include incorrect reasoning about relative positions (for example, which side of the frame an object is on), confusion between nearby objects, and object hallucination where the model mentions or reasons about an object that does not appear in the video. These cases indicate that, although keyframes provide strong appearance anchors, the current model sometimes struggles to maintain precise spatial grounding when combined with sparse motion information.

Temporal comprehension errors. Temporal errors arise when the model fails to reason over longer time spans or across multiple scenes. Typical failure patterns include confusion about the order of events, misidentification of the stage of an activity, or inability to track how a situation evolves over time. For instance, the model may answer a question about a later scene using information from an earlier one, or conflate two visually similar but temporally distinct segments. This suggests that maintaining coherent temporal context across keyframes and codec-derived motion cues

remains a central challenge.

Motion comprehension errors. These errors correspond to failures in understanding local motions and short-term actions. Representative examples include questions about subtle gestures, small object manipulations, or fine-grained action transitions, where our model either misses the relevant motion cue or confuses similar actions. Although our model employs the RMR module that transforms block-level codec motion vectors into refined motion features aligned with dense optical flow, these cases indicate that the refined signals are still not sufficiently informative for certain fine-grained dynamics.

Annotation errors. This category includes samples where the ground-truth supervision itself is unreliable. Typical examples involve incorrect or inconsistent annotations, as well as vague questions that admit multiple plausible answers. In such cases, our model's prediction is reasonable given the video content but is still counted as an error because it does not match the provided label.

Overall, Table 11 shows that temporal and spatial comprehension errors are more frequent than pure motion comprehension errors within the subset of cases where our model fails. Despite our model's stronger overall performance and qualitative behavior, these failure modes indicate that our model still faces challenges, not only in the quality of local motion cues, but also in how temporal and spatial features are aggregated over keyframes.

G. Scene-aware Video Preprocessing

Fig. 5 shows an example of the proposed scene-aware video preprocessing and the corresponding motion vectors extracted from the H.264 codec. Each row shows a sequence of frames, where scene-adaptive I-frames (e.g., Frames 0 and 18) provide appearance keyframes and the intervening P/B-frames are represented by block-wise motion vectors overlaid on the RGB images. The sequence begins with a top-down view of a bowl and then transitions to a frontal view of the person cooking; this scene change is captured by inserting a new I-frame at Frame 9, while frames within the same scene share a stable background and exhibit locally coherent motion patterns.

The overlaid motion vectors form coarse but informative motion fields. Large vectors concentrate around the hands, utensils, and ingredients, while the background remains mostly static. Although codec motion vectors are block-based and sparse, they approximate the underlying optical flow by indicating the direction and magnitude of local displacements between consecutive frames. The model interprets these vectors as pseudo optical flow and uses them to encode how objects and body parts move over time, while I-frames supply high-quality appearance cues at key timestamps. The RMR module further refines these codec-derived motion cues by mapping them to dense optical flow targets, yielding smoother and more temporally consistent motion representations. This preprocessing step therefore provides ReMoRa with scene-aware spatio–temporal inputs that capture both scene changes and fine-grained motions at a low computational cost.

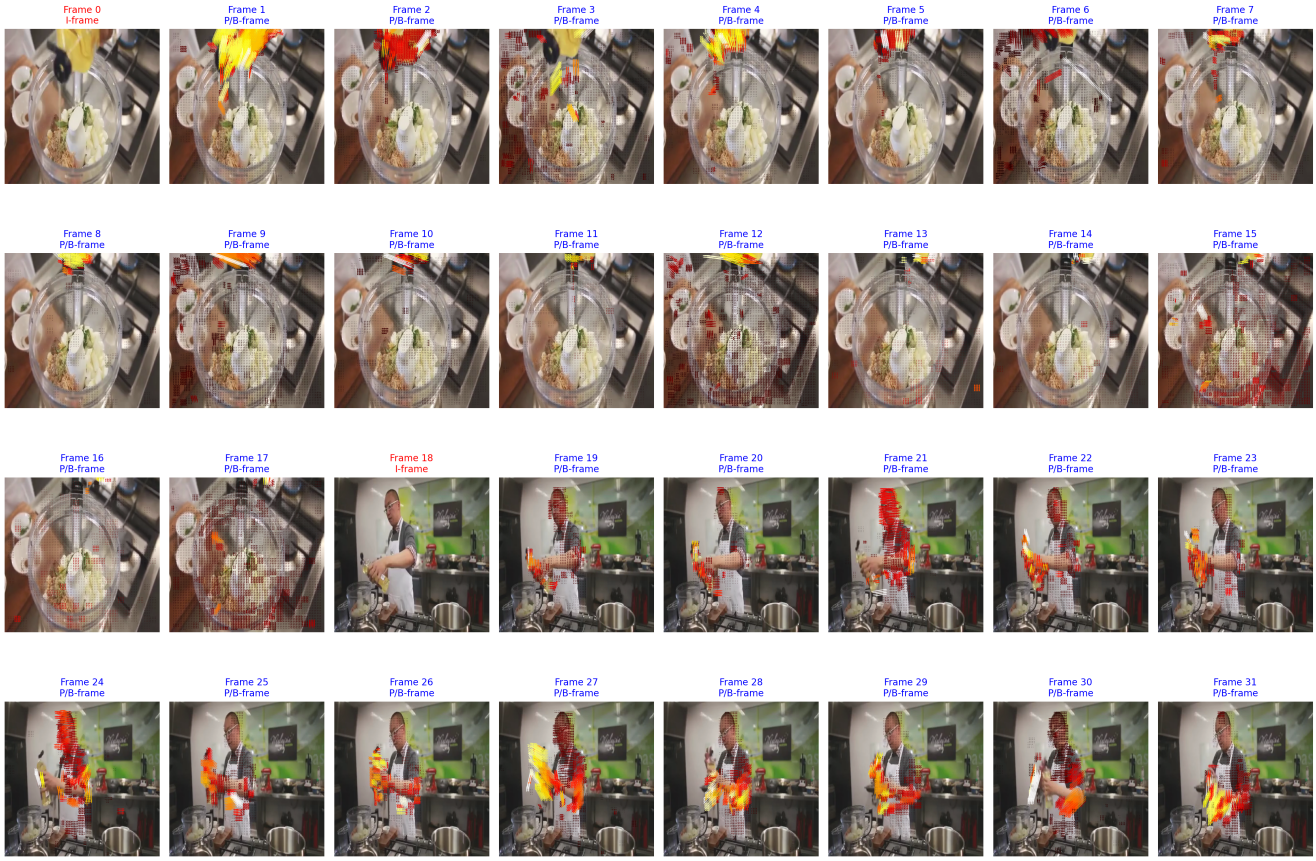


Figure 5. Example of scene-aware video preprocessing. Frames 0 and 18 are scene-adaptive I-frames used as keyframes, and the remaining frames are P/B-frames with overlaid codec motion vectors.